

OS/386: EMULATING DOS FOR PROTECTED MODE

For the PC Systems Integrator

September 1988 • \$3.95 USA (Canada \$4.95)

MicroSystems

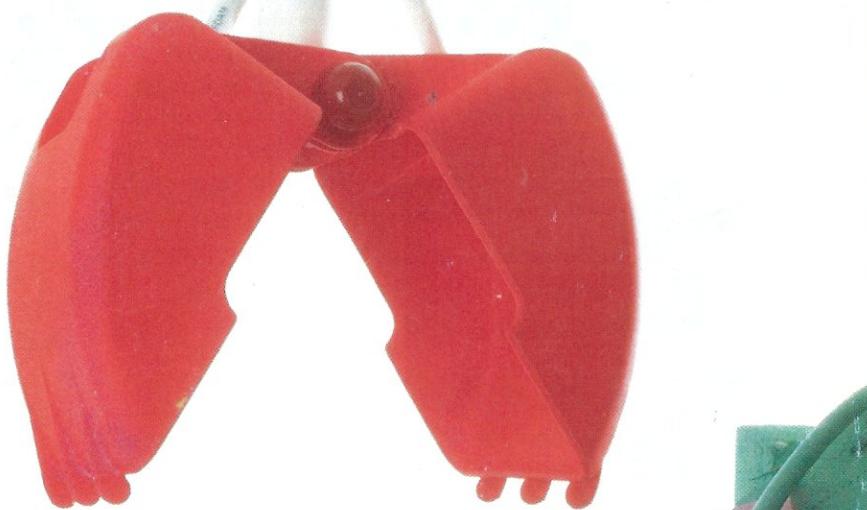
JOURNAL

Picking A NETWORK DATABASE

Six LAN
Databases
Compared

An SQL
Primer

SQL and Its
Network
Alternatives



***5-DIGIT 94043

0 38351

Make your programs millions of times smarter.

More and more, programmers and workstation builders are using DESQview 2.0 as a development tool. The reason is simple. They can create powerful, multitasking solutions today for the millions of DOS PCs in use today. Solutions comparable to those promised for tomorrow by OS/2.

The API Advantage

Programmers who take advantage of DESQview's API (Application Program Interface) get access to the powerful capabilities built into DESQview—multitasking, windowing, intertask communications, mailboxes, shared programs, memory management, mousing, data transfer, menu-building and context sensitive help.

Bells and Whistles

A program taking advantage of the DESQview 2.0 API can spawn subtasks for performing background operations or new processes for loading and running other programs concurrently. It can schedule processing after an interval or at a certain time. It can use DESQview's intertask communications to rapidly exchange data between programs, share common code and data; or interrupt at critical events. It can use DESQview's menuing and mousing capabilities to create menus. And there's lots more it can do.

Some of the applications under development right now using DESQview 2.0 API Tools: CAD, Medical systems, insurance, 3270 mainframe communications, network management, real estate, typesetting, point of sale, education, commodity trading, stock trading and online voting.

80386 Power

80386 programmers can take advantage of the 80386's protected mode for large programs, yet run on DOS and multitask in DESQview—side by side with other 80386 and DOS programs. The breakthroughs that make this possible: DOS Extenders from PharLap Software and AI Architects and DESQview support of these DOS extenders.

DESQview Developer Conference

So if you are a developer, looking to create programs with mainframe capabilities, but wanting to sell into the existing base of millions of DOS PCs, come to Quarterdeck's first DESQview API Developers Conference, August 16-18, 1988 at the Marina Beach Hotel, in Marina del Rey, California. For more information call or write us.

Come learn about the DESQview 2.0 API and 80386 DOS Extenders. Meet 80386 experts as well as those smart people who are creating DESQview 2.0 API workstations solutions.

And if you want to get a leg up before the conference, ask us about the DESQview API Tools for assembler or C programmers.

Bringing New Power to DOS. DESQview 2.0 API Toolkit.

The logo for Quarterdeck, featuring the company name in a stylized, bold, red font. To the left of the text is a graphic element consisting of a series of parallel, slanted lines of varying lengths, creating a sense of depth or a stylized 'Q' shape.

Quarterdeck Office Systems 150 Pico Blvd., Santa Monica, CA 90405
(213) 392 9851

Get your work done before 1991.

The future of personal computing is clear. More powerful PCs. Easier to use PCs. With graphics and character-based programs working side by side. Talking to each other. Multitasking. Windowing. Mousing. Getting your work done easier and faster.

Have it all now.

DESQview™ is the operating environment that gives DOS the capabilities of OS/2.™ And it lets you, with your trusty 8088, 8086, 80286, or 80386 PC, leap to the productivity of the next generation. For not much money. And without throwing out your favorite software.

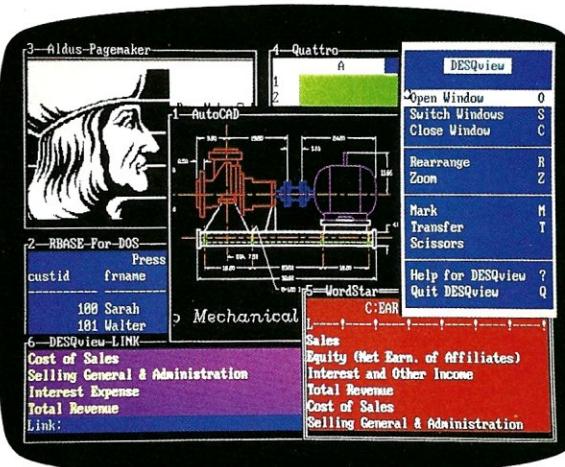
Add DESQview to your PC and it quickly finds your programs and lists them on menus. So you can just point to the program, using keyboard or mouse, to start it up. DESQview knows where that program lives. And what command loads it.

For those who have trouble remembering DOS commands, it adds menus to DOS. It even lets you sort your files and mark specific files to be copied, backed-up, or deleted—all without having to leave the program you're in.

Best of all, DESQview accomplishes all this with a substantial speed advantage over any alternative environment.

Multitask beyond 640K.

When you want to use several programs together, you don't have to leave your current program. Just open the next program. View your programs in windows or



full screen. Open more programs than you have memory for. And multitask them. In 640K. Or if you own a special EMS 4.0 or EEMS memory board, or a 386 PC, DESQview lets you break through the DOS 640K barrier for multitasking. If you have other non-EMS memory expansion products like AST's Advantage or the IBM® Memory Expansion Option, we have a

solution for you, too. The ALL CHARGE CARD™ 'unifies' all your memory to provide up to 16 megabytes of continuous workspace. DESQview lets you use this memory to enhance your productivity. You can start 1-2-3 calculating and tell Paradox to print mailing

lists. Open more programs than you have memory for. And multitask them. In 640K. Or if you own a special EMS 4.0 or EEMS memory board, or a 386 PC, DESQview lets you break through the DOS 640K barrier for multitasking. If you have other non-EMS memory expansion products like AST's Advantage or the IBM® Memory Expansion Option, we have a

labels while you're writing a report in Word Perfect, or laying out a newsletter in Ventura Publisher, or designing a building in AutoCAD.

DESQview even lets you transfer text, numbers, and fields of information between programs.

Fulfill the 386 promise.

For 8086 PC users, DESQview becomes a 386 control program when used in conjunction with Quarterdeck's Expanded Memory Manager (QEMM)-386—giving faster multitasking as well as virtual windowing support.

And when you use DESQview on an IBM PS/2™ Model 50 or 60 with QEMM-50/60 and the IBM Memory Expansion Option, DESQview gives you multitasking beyond 640K.

Experts are voting for DESQview. And over a million users, too.

If all of this sounds like promises you've been hearing for future systems, then you can understand why over a million users have



chosen DESQview. And why PC Magazine gave DESQview its Editor's Choice Award for 'The Best Alternative to OS/2,' why readers of InfoWorld twice voted DESQview 'Product of the Year'

why, by popular vote at Comdex Fall for two years in a row, DESQview was voted 'Best PC Environment' in PC Tech Journal's Systems Builder Contest.

DESQview lets you have it all now.

YES!
I need increased productivity now!

Name _____

Address _____

City _____ State _____ Zip _____

Payment Method Visa MasterCard Expiration _____

Account #

Qty	Product	Format	Price Each	Totals
	DESQview 2.0	<input type="checkbox"/> 5-1/4 <input type="checkbox"/> 3-1/2	\$129.95	
	QEMM-386	<input type="checkbox"/> 5-1/4 <input type="checkbox"/> 3-1/2	\$59.95	
	QEMM-50/60	<input type="checkbox"/> 5-1/4 <input type="checkbox"/> 3-1/2	\$59.95	
	ALL CHARGE CARD (Special for DESQview owners)		\$200.00*	

Shipping & Handling \$5 in USA / \$10 outside USA

Calif Residents add 6.5%

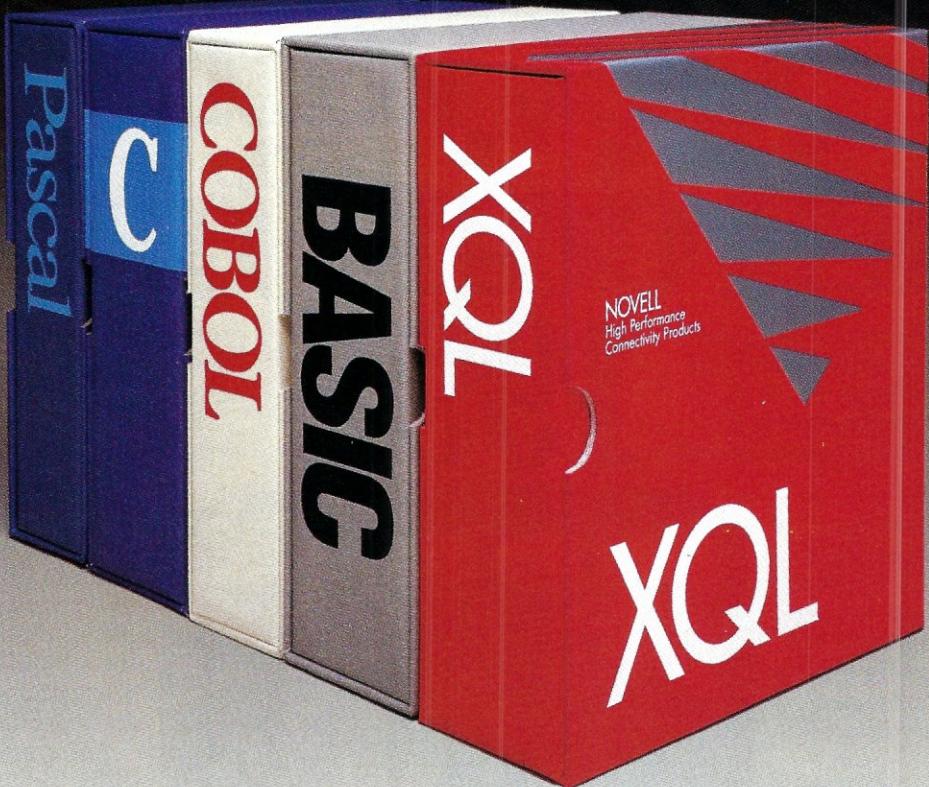
Grand Total

Quarterdeck

150 Pico Boulevard, Santa Monica, CA 90405
(213) 392-9851

DESQVIEW SYSTEM REQUIREMENTS:
IBM Personal Computer and 100% compatibles (with 8086, 8088, 80286, or 80386 processors) with monochrome or color display; IBM Personal System/2 Memory: 640K recommended; for DESQview itself 0-145K+ Expanded Memory (Optional); expanded memory boards compatible with the Intel AboveBoard; enhanced expanded memory boards compatible with the AST RAMpage; EMS 4.0 expanded memory boards; Disk: two diskette drives or one diskette drive and a hard disk; Graphics Card (Optional): Hercules, IBM Color/Graphics (CGA), IBM Enhanced Graphics (EGA), IBM Personal System/2 Advanced Graphics (VGA); Mouse (Optional): Mouse Systems, Microsoft and compatibles; Modem for Auto-Dialer (Optional): Hayes or compatible; Operating System: PC-DOS 2.0-3.3; MS-DOS 2.0-3.2; Software: Most PC-DOS and MS-DOS application programs; programs specific to Microsoft Windows 1.0-2.03, GEM 1.1-3.0, IBM TopView 1.1; Media: DESQview 2.0 is available on either 5-1/4" or 3-1/2" floppy diskette.

*This ALL CHARGE CARD is designed for the IBM PC AT and PS/2 250 and 60. If you have another type of 80286-based PC, there's a version for you, too. Please call 1-800 387-2744 for special ordering information. Offer expires August 31, 1988. Trademarks are property of their respective holders: IBM, OS/2, PS/2, 1-2-3, Paradox, Word Perfect, Ventura Publisher, AutoCAD, Intel, Above Board, AST, RAMpage, Advantage, Hercules, Mouse Systems, Hayes, Microsoft, Windows, TopView.



Finally, there's an SQL that gets back to BASIC. And COBOL. And C. And Pascal.

As a programmer, you've probably already faced it—the database dilemma. Do you use an SQL for easy database handling, or a true programming language for maximum power and flexibility?

Now you can do both with XQL®, the relational data management system from the developers of Btrieve.®

The Programmer's SQL. With XQL, you can access your data with the ease of Structured Query Language through simple subroutine calls from traditional programming languages. XQL supports standard SQL syntax, including subqueries, unions and security groups.

XQL Relational Primitive Operations. In addition, XQL lets you bypass the SQL level and perform highly efficient, relational primitive operations directly. You get all the functionality of a relational database model without the constraints of a 4th generation language.

Building on Btrieve. The heart of Novell's family of data management tools is Btrieve. By letting you access multiple records at a time, XQL adds a powerful dimension to Btrieve. XQL incorporates sophisticated data manipulation features which

allow you to access data by field name, move forward or backwards through the database, compute fields from other fields or constants, and even work with composite records built from multiple, joined Btrieve files.

Like Btrieve, XQL offers features like multi-user support, fault tolerance, comprehensive documentation, and expert technical support. And you never pay royalties on your XQL applications.

Solve the database dilemma with XQL, the SQL that speaks your language. Only \$795.* See your Authorized Novell Gold Reseller, or call us at (512) 346-8380.

For more information, call from your modem 1-800-444-4472 (8 bit, no parity, 1 stop bit) and enter the access code NVXQL13.



For software solutions,
you should be seeing red.

*Suggested retail price (US dollars) ©1988 Novell Inc., World Headquarters, 122 East 1700 South, Provo, Utah 84601 (801) 379-5900
Requires Btrieve 4.x and PC-DOS or MS-DOS 2.x, 3.x.

FEATURE ARTICLES

What To Look for in a LAN Database

Deciding which database has the data processing power you need for your network depends on a number of factors. When it comes to choosing LAN database software, one size does not fit all.

by *Patrick H. Corrigan* 18

An Introduction to SQL

What exactly is Structured Query Language? And is it the hope for the future of databases? This primer explains the ins and outs of SQL databases for the PC.

by *Edward Dowgiallo* 24

Structured Query Language and Its LAN Alternatives

SQL is now available for local area networks. As this comparison of competing database software illustrates, SQL is only one viable approach to database strategies for the LAN.

by *William Wong* 38

Using Btrieve with Turbo C and Turbo Pascal

There are a host of tools available to facilitate LAN database programming. In this article, the authors explain why they use Btrieve as their programming tool of choice to create databases for Novell NetWare.

by *Ed Rought and Tom Hoops* 54

PRODUCT REVIEWS

A.I. Architects' OS/386

This article provides a close look at this 386 operating system that serves as a platform for 32-bit real- and protected-mode DOS programming.

by *Howard Vigorita* 46

COLUMNS

From the Publisher's Desk by *Laird Foshay*

Uncontrolled Intersection Ahead: PC Databases 4

The C Forum by *Don Libes*

Understanding registers 8

LANscape by *Patrick H. Corrigan*

Twisted-Pair Networking Part II: The Wiring Closet 14

Database Queries by *P.L. Olympia, Ph.D.*

Report Writing in dBASE IV 62

Scientific Computer User by *A.G.W. Cameron*

Should Scientific Computing Use C? 64

The UNIX File by *Phil Hughes*

Floppy Disk Data Storage and Transfers 66



About the cover: This month's cover depicts a game that all network managers have to play, "Pick Your Database." If you go to the computer midway, you can hear the barker shouting, "You pays your money and you takes your chances!" But when choosing a network database environment, you can't afford to take chances. That's why we have devoted this issue to explaining network databases, including an overview of emerging SQL technology, a comparative look at database packages for local area networks, and a discussion of using Btrieve to program network databases.

DEPARTMENTS

News & Views 6

Ad Index 56

New Products 70

Micro/Systems Journal (ISSN 8750-9482) is published monthly by M & T Publishing, Inc., 501 Galveston Drive, Redwood City, CA 94063; (415) 366-3600. Second-class postage paid at Redwood City and at additional entry points. **POSTMASTER:** send address changes (form 3547) to *Micro/Systems Journal*, Box 3713, Escondido, CA 92025. **Change of Address:** Please send old label and new address to *Micro/Systems Journal*, Box 3713, Escondido, CA 92025. **Customer Service:** For subscription orders and changes of address, call toll-free: (800) 321-3333; in California call (800) 331-4164; outside U.S. call (619) 485-9623. **Subscription Rates:** U.S., \$29.97 for one year, \$56.97 for two years. Foreign orders must be prepaid in U.S. funds with additional postage. Add \$10 per year for surface mail to all countries or add \$20 for airmail to Canada and Mexico; \$28 for airlift/surface to other countries. Entire contents copyright ©1988 by M & T Publishing, Inc., unless otherwise noted on specific articles. All rights reserved.

Uncontrolled Intersection Ahead: PC Databases

Database technology for the personal computer has reached a crossroads. If PC database programmers are not already working as systems integrators, they will soon feel the pressure to become more involved in hardware/software configuration decisions. The database-related choices that systems integrators face in the years ahead will both stretch their skills and provide great opportunities. For the uninformed, or the unlucky, deadly collisions and productivity gridlock could result.

dBase, the broad database standard of the first generation of personal computing, seems to be finally running out of gas. Not many technical professionals or market analysts expect dBase to remain dominant in the changing PC market. Despite the fact that IBM, Microsoft, Oracle, Lotus, and every other major player have focused efforts there, it is unlikely that any leader in the PC database market will own the field with more than a 50-percent market share, as Ashton-Tate has with dBase almost since its introduction in the days of CP/M. The reasons for this fundamental shift toward fragmentation in the PC database market can be summed up in one word: complexity.

PCs are no longer the simple productivity machines they appeared to be just a few short years ago. Although they still offer the individual user a significant tool to increase productivity in writing and budgeting, PCs have been co-opted by the organization to play a much broader role. They have become the platform of choice for company-wide systems. And these emerging PC systems are a vastly more difficult environment for any one software vendor or systems integrator to address than the relatively simple 640K, stand-alone PC.

Rather than simply remaining one of the big three applications along with word processing and spreadsheets, databases promise to form the cornerstone of a new generation of PC systems. They are the primary storage medium for customer lists, inventory records, and accounting data; a whole world of information that can be shared and manipulated by many users of personal computer systems. SQL promises to be a broad standard that will allow data to be shared across a wide variety of applications, which can also be spread over a network. Databases will form the foundation for a new generation of workgroup applications, including "smart" electronic mail and shared calendars. The real problem is getting this advanced software to run in the wildly heterogeneous world of PC systems.

The database crossroads ahead is, in fact, an almost hopelessly complex convergence of operating systems, networks, servers, processors, and "standards" from a dizzying array of vendors. The crossroads looks like a Boston rotary at rush hour, with traffic pouring in from all angles at frightening speeds, horns blaring. The comfortable world of DOS PCs and Macintoshes is giving way to a world where NetWare, OS/2, LAN Manager, and UNIX may all play significant roles. It is even conceivable that your customers or company will have all of these pieces of systems software running at the same site and want them all to talk to each other via database servers.

In this new world, the PC systems integrator —no matter what the job title—will play the pivotal role of arbiter and coordinator. You will have to sort out user needs and choose from a wide array of "standards" and vendors. Our goal here at *Micro/Systems* is to make your work a little easier by addressing the practical impact of these converging technologies. We'd like to provide you with the techniques and technical product knowledge to prepare you to meet the challenge ahead.



Laird Foshay,
Publisher

For the PC Systems Integrator
MicroSystems
JOURNAL

EDITORIAL

Founder and Editor	Sol Libes
Technical Editors	Stephen R. Davis Don Libes
Associate Editors	Lennie Libes Susan Libes
Contributing Editors	A.G.W. Cameron Patrick Corrigan P.L. Olympia
Managing Editor	Thomas M. Woolf

PRODUCTION

Art & Production	Larry L. Clay
Director	Joe Sikoryak
Art Director	Barbara Mautz
Asst. Art Director	Lorraine Buckland Margaret Anderson

CIRCULATION

Director of Circulation	Maureen Kaminski
Asst. Circulation Mgr.	Andrea Weingart
Fulfillment Coord.	Francesca Martin
Newsstand Mgr.	Sarah Frisbie

ADMINISTRATION

V.P. Finance & Operations	Kate Wheat
Business Mgr.	Betty Trickett
Accounting Supv.	Mayda Lopez-Quintana
Accts. Payable Asst.	Luanne Rocklewitz
Accts. Receivable Asst.	Wendy Ho

ADVERTISING

Advertising Director	Richard Mixter
National Account Mgr.	Dwight Schwab
National Account Mgr.	Tami Brenton
Advertising Coord.	Shaun Hooper

M&T Publishing, Inc.

Chairman of the Board	Otmar Weber
Director	C.F. Von Quadrat
President & Publisher	Laird Foshay
V.P. of Publishing	William P. Howard

Micro/Systems Journal (ISSN 8750-9482) is published monthly by M&T Publishing, Inc., 501 Galveston Drive, Redwood City, CA 94063; (415) 366-3600. Second-class postage paid at Redwood City and at additional entry points.

Article Submission: If you have a specific area of expertise or interest and would like to contribute, please write *Micro/Systems Journal*, P.O. Box 1192, Mountainside, NJ 07092; (201) 522-9347, or contact M&T Publishing, Inc., 501 Galveston Drive, Redwood City, CA 94063; (415) 366-3600. Please do not submit articles without first contacting the editors.

Correspondence: Please send letters to the editor to *Micro/Systems Journal*, 501 Galveston Drive, Redwood City, CA 94063. Other editorial correspondence may also be directed to P.O. Box 1192, Mountainside, NJ 07092. The editors may also be reached via MCI Mail (SLIBES or MSJ).

Advertising Rates: Available upon request. Call (415) 366-3600 or write to: Advertising Department, *Micro/Systems Journal*, 501 Galveston Drive, Redwood City, CA 94063.

Concurrent DOS 386

Think small in a big way

When you think multiuser/multitasking, think Concurrent™ DOS 386, the big name in small systems from Digital Research®, architects of the first standard operating system for personal computers. Now, Concurrent DOS 386 allows multiple users to share peripherals, files and applications, using serial terminal workstations linked by RS-232 cables to the system. It's fast, reliable and economical.

The big news today is small systems

Concurrent DOS 386 meets the increasing demands placed on small systems by supporting multiple DOS programs on both the system console and attached terminals. You can run popular programs such as Lotus® 1-2-3®, dBase® III, WordPerfect® and many more, with full math coprocessor support. The system runs up to 255 tasks simultaneously, with full intertask communications and byte-level record, file and device locking.

For people who hate waiting in line

Concurrent DOS 386 brings you all the remarkable speed and power of the Intel® 80386 processor. A prioritized preemptive scheduler allows task execution

and intertask communication by several users at near full processor speed while letting some tasks "interrupt" others according to the needs of each user.

A small system with a big memory

Concurrent DOS 386 gives you access to four gigabytes of linear physical memory. Its powerful memory paging capability fully supports the Expanded Memory Specification with no additional hardware or software.

Menus at a touch

Now you can create and customize menus, while programmable function keys let you condense complex commands to a single keystroke. The file manager runs standard operating system functions, plus you have an on-line help facility, text editor and support for DOS-based device drivers.

Multiuser color graphics

Now with the introduction of the newest member of the Concurrent DOS family, Concurrent DOS 386/Multiuser Graphics Edition, your demands for high-resolution EGA bit-mapped graphics in the workstation environment can

be met. Take advantage of advanced technology allowing you to run popular DOS-based graphics programs on individual workstations as well as on the system console without sacrificing system performance. Ask us about this exciting new version of Concurrent DOS 386.

All you have to remember is Concurrent DOS 386

Concurrent DOS 386 from Digital Research is the name to remember when it comes to 386 technology. The power and versatility of Concurrent DOS 386 are giving a new meaning to the word multiuser.

**CALL DIGITAL RESEARCH AT
1-800-443-4200 AND ASK FOR OUR
CONCURRENT DOS PROGRAMMER
INFORMATION KIT.**

CONCURRENT DOS 386: SHARING THE SYSTEM AFFORDABLY

Digital Research and the Digital Research logo are registered trademarks, and Concurrent is a trademark of Digital Research Inc. Other product names are registered trademarks or trademarks of their respective owners. Specifications are subject to change without notice. Copyright © 1988, Digital Research Inc. All rights reserved.

 **DIGITAL RESEARCH®**

CIRCLE 98 ON READER SERVICE CARD

News & Views

by Sol Libes

Random Rumors & Gossip

Prototypes of the next generation of portables are expected to be shown at the upcoming fall Comdex to be held in Las Vegas in November. Look for them to sport active matrix, high-contrast displays that are easily readable under fluorescent lighting, VGA controllers, and trackballs for use with Windows or OS/2.

IBM and AT&T are rumored about to adopt Adobe's Display PostScript (DP) screen format. Apollo, DEC, NeXT, and Hewlett-Packard are already on the Display PostScript bandwagon. The DP language makes graphics resolution device-independent and provides graphics software portability.

Borland is rumored to be readying "Turbo Assembler" as well as Version 5.0 of its popular Turbo Pascal and new versions of Turbo C and Turbo BASIC, all for release before year-end. They all will have debuggers and faster file I/O.

With 386-based systems gaining larger market share faster than expected, word is that **IBM** and **Microsoft** have moved their timetable for OS/2-386 forward. Originally scheduled for introduction late in 1989, they are now expected to begin releasing the product in pieces with the first part coming as early as the second quarter of next year. IBM is expected to shortly announce that it will offer System/3X applications to run under OS/2 on its 386-based PS/2 systems.

And there are rumors that **IBM** has decided to hold up releasing its 286- and 386-based laptops that have been out with beta testers since the beginning of the year. After previous disappointments in the portable and laptop marketplace, IBM appears to be hesitating for fear of making more marketing mistakes.

PS/2 Sales Below IBM Expectations

While 286- and 386-based AT clone makers are enjoying their best sales year ever, there are growing indications that sales of IBM PS/2s are lagging behind IBM's projections. First of all, IBM's complete line of PS/2 systems are being widely and heavily discounted by dealers, and a large number of PS/2 systems are now available in the gray market. Further,

IBM has been forced to offer a trade-in to purchasers of PS/2 systems. Owners of old IBM PC/AT machines are receiving as much as \$1,200 in trade for their old machines.

But the most significant indicator of disappointing PS/2 sales is IBM's decision to shut down PS/2 manufacturing at its Boca Raton, Florida, factory and transfer more than 1000 employees to other locations.

386SX-based Systems Arrive

Compaq Computer began shipping the first computer to use Intel's new 386SX processor chip in early July. NEC, IBM, and NCR are expected to release their systems shortly. The 386SX is a 16-bit I/O version of the 80386 and is expected to lower the cost of 386 systems by \$400 to \$700. These systems will be able to run 386-based software, such as Windows/386, Paradox 386, AT&T's UNIX System V/386, and a future version of OS/2-386.

However, these systems will be limited to a maximum of 16 MB of memory compared to a maximum of 4 gigabytes on 80386 systems (24-bit versus 32-bit address bus), and applications should execute about 10 to 30 percent more slowly because of the 16-bit I/O and slower clock rate.

Intel claims that more than 50 companies are developing systems using the 386SX chip. Early entries from AST Research, NEC, and Zenith are expected to be shown at Comdex in November, with production shipments expected early next year. No doubt most of these systems will be laptops or portables.

The introduction of these systems has already caused the prices of 286-based systems to drop. Compaq, for example, has reduced the price for its 286-based systems by about \$400.

OS/2-EE Communications Manager Alternatives

You have to give AST Research credit as a serious innovator. When Lotus/Microsoft/Intel announced their system to manage memory beyond 640K, AST introduced its own extended memory manager and cooperated with other companies that also wanted to use it. It was eventually recognized as the superior system and adopted

as the EMS 4.0 standard, which is now widely used by PC manufacturers.

When IBM introduced its PS/2 MicroChannel Architecture multiprocessor system, AST again matched them by adding the multiprocessor feature to the standard AT bus. And AST went one better by introducing a hard-disk controller that used the multiprocessor feature to provide faster throughput to the disk.

Now AST is at it again, revealing plans to offer a competitor to IBM's OS/2 Extended Edition Communications Manager (CM). Naturally, it plans to provide additional features over those provided in the IBM product. The company claims the product will work with all versions of OS/2.

Digital Communications Associates has also announced an alternative to IBM's CM called "Select CS." It will also offer features not found in the IBM product. The IBM CM has several significant gaps in functionality that AST and DCA have filled. IBM's initial release of the CM lacks emulation for the IBM 5250 terminal and will not run in a Presentation Manager window. The IBM CM also lacks support for LAN 3270 gateways and IBM's 3270 High-Level Language Application Program Interface (HLLAPI). No doubt these features will be provided in a future update, but initial versions of both the AST and DCA CMs will include these features.

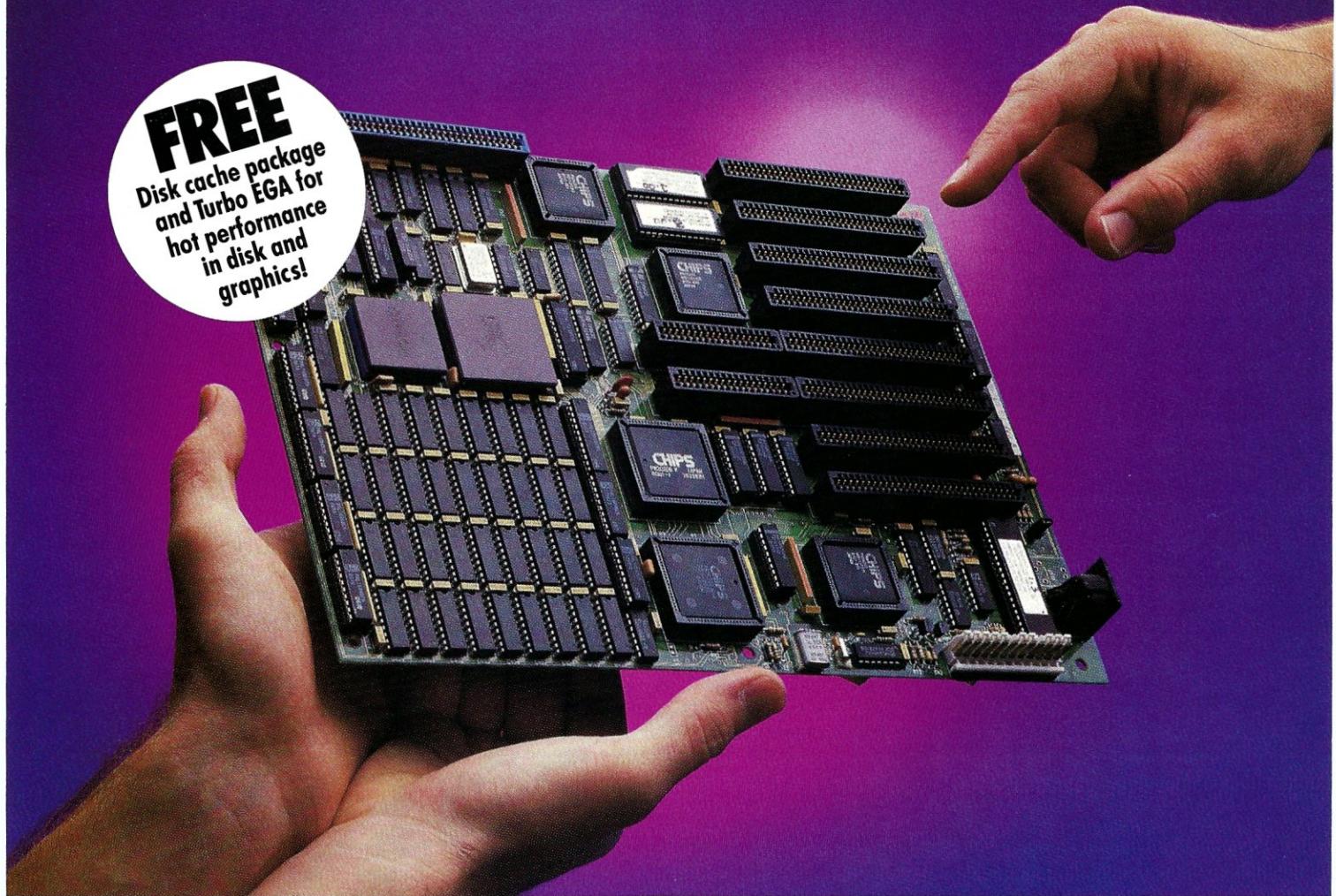
DCA and AST will also support a variety of IBM and non-IBM communications and network adapters while IBM will only support its own hardware.

Who Owns the PC CPU Market?

Mike Slater publishes a most interesting monthly newsletter called *Microprocessor Report* [call (415) 494-2677 for subscription information]. The following is extracted from a recent copy.

Intel controls the bulk of the high-end PC-compatible CPU market with its 386/387/82385 chip set (CPU, math coprocessor, and cache controller). "The sole-sourced Intel chip set costs well over \$1,000, even in volume, for the 20-MHz version." However, two new types of products are due next year that will compete with the 386 chip set, which should cause Intel to drop 386 prices significantly. First will be the introduction of RISC designs able to emulate the 386 at comparable speeds. Second will be the introduction of "binary recompilers" that will be used to recompile DOS applications to the "native" mode of another processor, thus enabling them to run even faster than on 386-based systems. These would most likely run under UNIX. □

FREE
Disk cache package
and Turbo EGA for
hot performance
in disk and
graphics!



CONVERT YOUR PC, XT OR AT INTO A HIGHER FORM OF LIFE!

386 MOTHERBOARDS FOR 386 SPEED

Don't let your PC give up the ghost—Hauppauge has just arrived with a new spark of life: the 386 MotherBoard.TM Far more advanced than an accelerator card, our line of MotherBoards grace your PC, PC/XT, PC/AT or compatible with speeds equal to the IBM PS2 Model 80. And faster. Because we've built in 1 Megabyte of high speed RAM and a 387 math coprocessor socket for speeds that make users humble with awe.

OS/2 Compatible. To ensure a long, fruitful life, our 386 MotherBoard is compatible with the PC/AT (BIOS and I/O)—so you can run the new generation of DOS, OS/2. Our Board also runs Windows/386, UNIX V and PC-MOS/386. For more power, you'll find 16-bit expansion slots that accommodate the latest I/O expansion cards. No 386 accelerator card gives you so much versatility. **Only our 386 MotherBoard gives your PC a future with unlimited possibilities!**

The Critics Applaud! *PC Magazine* awarded our Board "The Editor's Choice" for 386 Replacement Boards. *PC World* called it "the Upgrade Product of the Year."

Technical Features • 16 MHZ 80386 • 1 Megabyte of 100 nsec 4-way interleaved RAM • PC/AT compatible I/O and BIOS for support of OS/2 • Six 8-bit expansion slots • two 16-bit expansion slots (four on 386 MotherBoard/AT) • One 32-bit expansion slot for up to 12 Megabytes of high speed memory • Battery-powered clock calendar

386 MotherBoard/PC or MotherBoard/XT	\$1495
386 MotherBoard/AT	\$1595
32-bit RAM Board	
(2 Mbytes installed; up to 4 Mbytes)	\$795
16 MHz 80387 math coprocessor	\$695
16-bit combination hard disk/ floppy disk controller	\$245

For more information on our easy-to-install MotherBoard, call: **1 (800) 443-6284**. In New York, call **(516) 434-1600**.

Hauppauge Computer Works, Inc.
175 Commerce Drive,
Hauppauge, New York 11788

Hauppauge!

by Don Libes

Understanding *registers*

The *register* keyword is used to indicate to the compiler that a variable should be stored in a register if possible. The idea is that the programmer is telling the compiler that a variable is frequently used, and placing it in a register will speed up the program.

This sounds pretty intuitive, and the idea really is. But there are non-intuitive points to be aware of, and that is what this column is all about. For example, it is a common belief that good compilers obviate the need for *register* declarations. This simply isn't true, as I will explain later. To get the best performance from your code, you must be prepared to use *register* declarations.

register declarations are used during code generation. At that time, the compiler decides what machine instructions will be used to execute your program. The *register* keyword requests that the variable be stored in a machine register. However, this may not always be possible. One obvious reason is that the machine may not even use registers. For example, stack-oriented machines, such as the Pyramid, don't have registers.

Compilers are free to ignore *register* declarations. Naturally, the Pyramid's C compiler does exactly that.

However, even register-oriented machines can ignore *register* in a declaration. For example, suppose you have a machine with eight registers and you declare nine register variables. Obviously, the compiler is not going to be able to satisfy your request. Right? Not necessarily, as we will see later.

In fact, most compilers will not allow the user to dictate the allocation of all the registers of the machine. For example, one register each is typically reserved for the stack pointer and the frame pointer (or activation

Don Libes is a computer scientist working on artificial intelligence in robot control systems in the Washington, D.C., area. He has also recently completed Life with UNIX, published by Prentice-Hall.

record). A couple of registers are generally kept for temporary computations. If you need to allocate and manipulate the registers more efficiently than the compiler, you are going to have to resort to assembler.

Well then, *how many* registers can we declare with any assurance? If your code is going to be ported to other architectures, the guaranteed answer is zero. Otherwise, you will find the answer only in your particular compiler's manual. This is precisely what is stated by the ANSI C standard.

If the documentation doesn't say — and this is not uncommon — then you can experiment by declaring a lot of register variables and examining the assembler output to see how many are manipulated directly as registers.

A simple program to do this may not always be possible. Some compilers perform *live range analysis*. This technique moves values out of registers when they are no longer being accessed, even though they are still in the current scope.

```
register int i, j;
/* no use of j in loop */
for (i=0; . . . ; i++) {
    . . .
}
/* no further use of i */
for (j=0; . . . ; j++) {
    . . .
}
```

For example, some compilers can create code that uses only one register for controlling both loops instead of two. Since use of *i* and *j* don't overlap, it is easy to share the register.

This is a particularly easy case, since *i* and *j* could actually share the same storage whether they were kept in registers or not. That is, if we literally substituted *i* for *j* throughout the function, it would still work correctly. This is often not possible, even when register sharing is.

```
for (i=0; . . .
for (j=0; . . .
for (i=0; . . .
```

In this piece of code, we see that *i* is used throughout, but the compiler can save it before entering the middle loop and restore it afterwards. This, again, allows *i* and *j* to use the same register with a small amount of overhead. Of course, you will have to experiment to see if your compiler can do this.

Good compilers will do it automatically (without the *register* declaration), but my experience is that these are not common. In any case, complicated expressions are even more unlikely to be optimally handled. It is to your benefit to declare as *register* the variables that are heavily used.

The most obvious problem is that compilers have no way to tell how many times loops are going to be executed. Bounded loops like the ones above are easy, but analysis is generally impossible for unbounded loops like the following:

```
while (1) {
    if (expr) break;
    . . .
}
```

In the face of these, the compiler cannot justify saving and restoring registers when there is no guarantee that it will save time during execution. To combat this, people have proposed language extensions to indicate primary flow of control, such as which branch of an *if* is more likely to be taken. However, these have not been adopted into the mainstream of C.

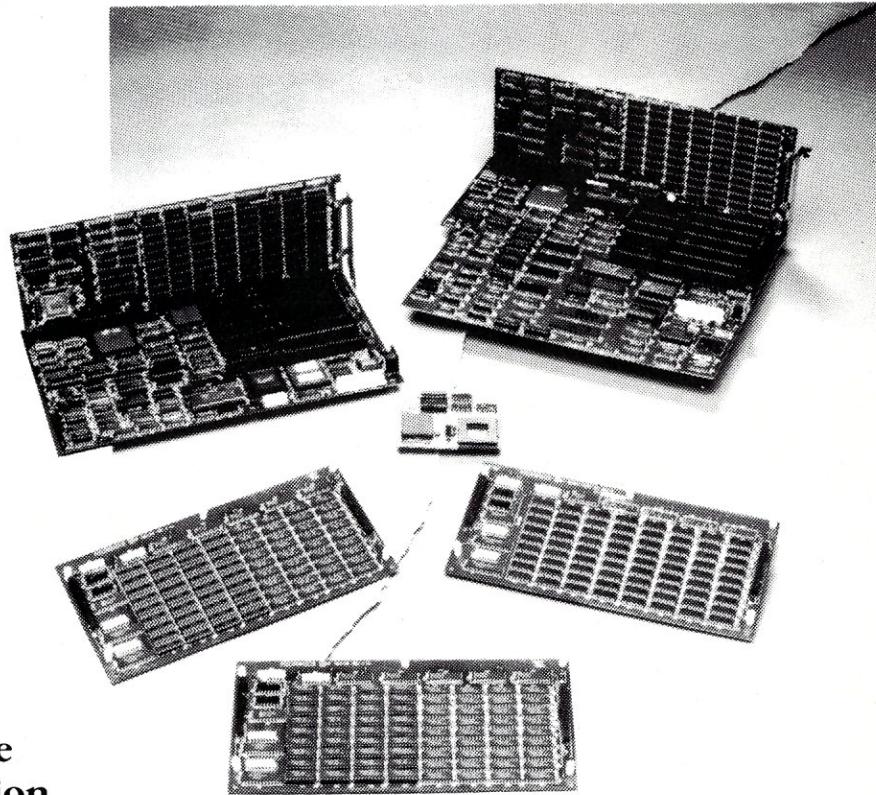
The only solution is to make judicious use of the *register* declaration yourself. Don't hold out for optimizing compilers to produce optimal code — that's not what they do. They just produce better code. You still have a responsibility to help the compiler generate the best code possible. A general rule of thumb is that if a variable is accessed more than four times, you should consider declaring it a register. Time your code with and without *register* declarations to see how *four* feels on your system.

Some people advocate using only one or two register declarations in a function, but I believe you should use as many register declarations as necessary. Even if your current compiler only uses the first two of them, you will get improvement in your code, since you are telling the compiler what is heavily used rather than having it guess.

Always make sure that you declare register variables in decreasing order of importance. When the compiler allocates registers to variables, it will follow your order of declarations. If it runs out of registers before getting

*Insist
On The Best
Micronics
Motherboards*

386



**uality,
Performance
and Innovation**

best describe our 80386 based board level product line. Now with both **AT** and **Baby size** and high speed **CACHE** memory.

Micronics is the leading supplier to **OEMs**, **VARs** and Systems Integrators that require the **best in 80386 technology**.

For a distributor near you, call
800 / 234-4386.

**Innovation
and
Performance**

MICRONICS
COMPUTERS INC.

© Copyright 1988 Micronics Computers, Inc.

Get a window on the future . . .



without giving up the present

At last, you can have the benefits of UNIX™ V.3 and high resolution graphics through X-Windows—without sacrificing the investment you've made in your DOS based 80386™ system.

The Easy Way to X-Windows

With the XSUBSYSTEM™ from Interfirm Graphic Systems, you'll have everything you need to convert your 80386-based machine into a powerful UNIX-based graphics workstation. Suddenly, you'll have all of the advantages of a multi-user, multi-tasking system—plus the incredible convenience of DOS programs running side-by-side with UNIX.

The XSUBSYSTEM Includes:

- 19" High resolution color or monochrome display
- Graphics controller
- UNIX System V.3 from Interactive Systems Corp.
- The X Window System software
- Optional hard disk with all software pre-installed
- Optional TCP/IP Ethernet controller and software

The Future of 386 Based Graphics

The XSUBSYSTEM gives you the best of both worlds - you get to keep both your hardware and your DOS software while you upgrade to a UNIX V.3/X-Windows System.

For example, with Interfirm's XSUBSYSTEM, you can take data created in a DOS application and paste it into a document that you are creating with the Frame Maker™ desktop publishing system.

It's the most sensible way we know to give your 386 based system the power of a true graphics workstation - at a sliver of the price.

At Interfirm Graphic Systems, we are pioneering UNIX-based high-performance graphics, utilizing the X Window System and other emerging non-proprietary standards. When you look to the future of desktop computing, look to Interfirm.



**INTERFIRM
GRAPHIC SYSTEMS**

3940 Freedom Circle
Santa Clara, CA 95054
(408) 970-7070
800-338-4513

Trademarks: UNIX-Bell Laboratories; XSUBSYSTEM - Interfirm Graphic Systems; MS DOS - Microsoft; X Window System - M.I.T.; 80386-Intel Corp; Frame Maker - Frame Technology Corp.

CIRCLE 109 ON READER SERVICE CARD

to the end of your list, you may have register variables that are not stored in registers. Make sure you can live with this.

Last month's column presented a function, *real_fast_strcpy*, that declared six *register* variables. This is more than most compilers will allow you to allocate, but they were arranged in descending importance. The ones at the end were used in loops that would only loop a few times, while the ones in the beginning were used in loops that would usually execute many times.

Using register chars usually causes the compiler to simulate small registers.

Addressing Register Variables

Do not use *register* on variables for which you intend to take the address. For example, look at the following code fragment:

```
int *i;  
register j;  
  
i = &j;
```

Pre-ANSI C compilers allowed such code. While the compiler executed the code, you would not get the speed-up you expected. The problem is that most machines do not have addressable registers. If a compiler saw such a code fragment, it could do two things:

1. ignore *register*, or;
2. generate extra instructions to move the value between register and memory whenever necessary.

Needless to say, implementors almost always took the first option. ANSI C has outlawed such constructs, and the compiler must complain. You may or may not agree with this, depending upon your position. Personally, I think this was a mistake, since it precludes certain optimizations from a class (admittedly small) of machines that can address registers. Furthermore, such a warning is more appropriate coming from *lint*. (I also believe you should run *lint* very often during program development.)

Register Parameters

Last month's column went into some detail on the use of register declarations on function parameters. I suggested that you avoid using *register* when declaring function parameters if this was to be portable code, since many compilers will ignore them. The best way to get the same effect is to declare new variables as *register* inside the body of the function and then assign them values from the parameters. You will not lose any speed, even if the compiler does correctly handle *register* parameters.

ANSI C function prototypes ignore the use of *register* declarations entirely. So, for example, the following

two statements declaring *strcpy()* are equivalent.

```
char *strcpy(char *dest,  
            char *src);  
char *strcpy(register char *dest,  
            register char *src);
```

These lines are not the function definitions, just declarations. Use of *register* is noted in function definitions. The rationale is that function prototypes are used to check that parameter types match, but *register* does not change the parameter type. If anything, this helps separate specification from implementation.

By the way, in case it isn't obvious, you cannot use *register* on variables

SERIOUS DEBUGGING AT A REASONABLE PRICE



All the speed and power of a hardware-assisted debugger at a software price

Hardware-level break points

REAL-TIME break points on memory locations, memory ranges, execution, I/O ports, hardware and software interrupts. More powerful break points than ANY software-only debugger on the market.

Break out of hung programs

With a keystroke - no external switch necessary. Even with interrupts disabled.

Breaks the 640K barrier

Soft-ICE uses ZERO bytes of memory in the first 1MB of address space.

Works with your favorite debugger

Soft-ICE can be used as a stand-alone debugger or it can add its powerful break points to the software debugger you already use.

Solve tough systems problems too

Soft-ICE is ideal for debugging TSRs, interrupt handlers, self booting programs, DOS loadable device drivers, non-DOS operating systems and debugging within DOS & BIOS. Soft-ICE is also great for firmware development because Soft-ICE's break points work in ROM.

How Soft-ICE Works

Soft-ICE uses the power of the 80386 to surround your program in a virtual machine. This gives you complete control of the DOS environment, while Soft-ICE runs safely in protected mode. Soft-ICE uses 80386 protected mode features, such as paging, I/O privilege level, and break point registers, to provide real-time hardware-level break points.

Buy Both and SAVE \$86

MagicCV	\$199
Soft-ICE	\$386

(603) 888 - 2386

Nu-Mega Technologies

PO Box 7607
Nashua, NH 03060-7607

30 day money-back guarantee
Visa and Master Card accepted

Both require 80386 PS/2 or AT compatible

Using Soft-ICE with CodeView gives you the features necessary for professional level systems debugging. MagicCV and Soft-ICE can work in concert with CodeView to provide the most powerful debugging platform you will find anywhere.

As an extra bonus, by ordering both MagicCV and Soft-ICE together you SAVE \$86.

Codeview is a trademark of Microsoft Corp.

RUN CODEVIEW IN ONLY 8K TM



CodeView is a great integrated debugger, but it uses over 200K of conventional memory. MagicCV uses advanced features of the 80386 microprocessor to run CodeView in a separate virtual machine in extended memory. This allows MagicCV to run CodeView using less than 8K of conventional memory on your 80386 PC.

Don't let 640K be your limit!

If you are closing in on the 640K limit and would like the power of CodeView, MagicCV is for you.

Find those hidden bugs!

Even if you're not closing in on the 640K limit, running CodeView with MagicCV makes your debugging environment much closer to the end user's program environment. You can use CodeView to locate subtle bugs that only occur when there is plenty of free memory.

MagicCV is easy to use

If you are a CodeView user, you already know how to use MagicCV too. Just type MCV instead of CV, everything else is automatic.

MagicCV with Soft-ICE

"It is a unique, effective solution to a wide range of critical debugging problems"

COMPUTER LANGUAGE -- April 1988

CIRCLE 103 ON READER SERVICE CARD

declared *extern*. Since all variables declared outside of any function are implicitly *extern*, this means you can only use *register* inside of a function.

Register types

If you say *register foo, foo* is declared to be of type *int*. *register* can also be applied to other types, although such use is generally not portable. You will have to consult the reference manual for your C compiler to be sure.

Most compilers will generate code for *register short* and *register char*, since these will fit in an *int*. However, keep in mind that the natural word size of a machine is an *int*, and registers are generally *int*-sized. Using *reg-*

ister chars, for example, usually causes the compiler to simulate small registers by performing some of the computations in registers and some in memory. For example, a series of additions may normally cause overflow in memory, but not when done by *int* register computations. In order to preserve semantics, the compiler has to generate extra instructions.

It is better for the programmer to be aware of this and write the code using *int* registers, then expect the compiler to work around the problem. Any type variable can be declared *register*. However, larger types such as *structs* may not see any improvement, either because they are

too large for a register, or the compiler writer chose to ignore the use of *register* on anything but *ints*. This is not uncommon.

register floating point variables are more likely to be supported, since many machines have floating point registers. These can be used directly by applying *register* to the declaration. Such usage tends to be less portable, if only because your code may someday run on a machine that has no floating point registers, and your request for registers will conflict with requests for the *int* registers.

setjmp/longjmp

In a past *C Forum* (*Micro/Systems*, January 1988), I discussed *setjmp* and *longjmp* in detail. I made two important points in that column that are relevant here. (The second was actually incorrectly typeset, and this is my opportunity to state it correctly.)

1. *longjmp* only restores register variables, but not any others when the saved environment is restored.
2. There is no guarantee that the variables declared as *register* will have their values saved by *setjmp*, since there is no guarantee that the compiler will actually put your variables in a register because *register* is only a hint to the compiler.

The second point is extremely pertinent when trying to create portable code. If you need more information on either of these, please refer back to the earlier issue.

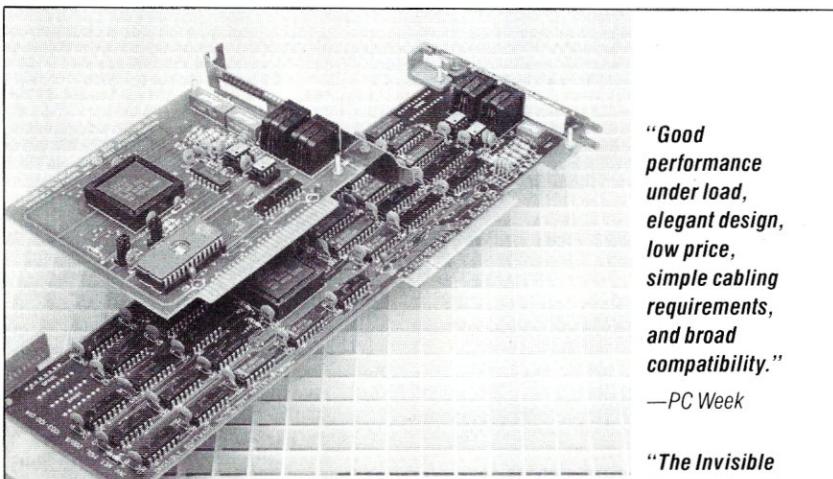
Conclusion

Most people think that the good thing about *register* is that if it is misused, it will be ignored. This simply isn't true. At best, misuse of *register* can cause your program to execute more slowly than without it. At worst, *register* can cause great confusion due to subtle interactions with *setjmp* and *longjmp*.

On the other hand, *register* declarations are great when used correctly. By allowing you to give hints to the compiler during code generation, you can get better performance than would be possible even from the best optimizing compilers without *register* declarations. □

All the source code for articles published in *Micro/Systems* is available on an MS-DOS disk. To order, send \$14.95 to *Micro/System Journal*, 501 Galveston Drive, Redwood City, CA 94063. Please specify the issue number.

Did you find this article particularly useful? Circle number 1 on the reader service card.



Best value in a complete LAN New! Invisible Network.

Best value. New Invisible Network™ from Invisible Software is a *real* LAN. It transfers data at 1.8Mbps. In one high performance package, it combines power, reliability, ease of installation, and low cost. We have included all the hardware and software. And backed it with a lifetime limited warranty plus a 30-day money-back guarantee.

Broad compatibility. Invisible Network includes the NET/30™ Network Operating System at no additional expense. Invisible Network is also fully compatible with the IBM NETBIOS standard, the IBM-PC LAN Program™ and Novell Advanced Netware™. So you can run all the most popular network applications.

Installs simply. You easily install the LAN yourself by inserting an interface card (with built-in modular jack) into each network computer. It's that simple.

"Good performance under load, elegant design, low price, simple cabling requirements, and broad compatibility."

—PC Week

"The Invisible Network is a darn good deal."

—PC Magazine

"A joy to use. I highly recommend it."

—Data Based Advisor



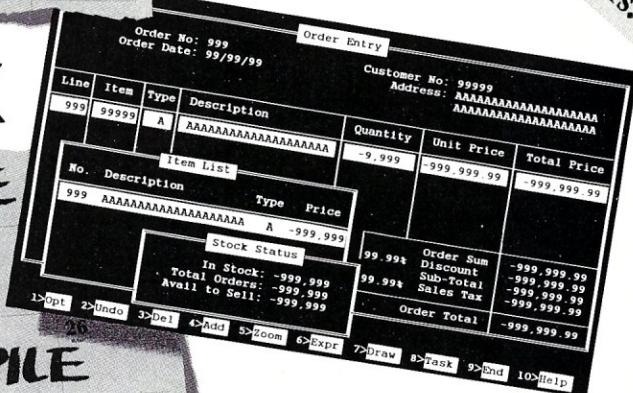
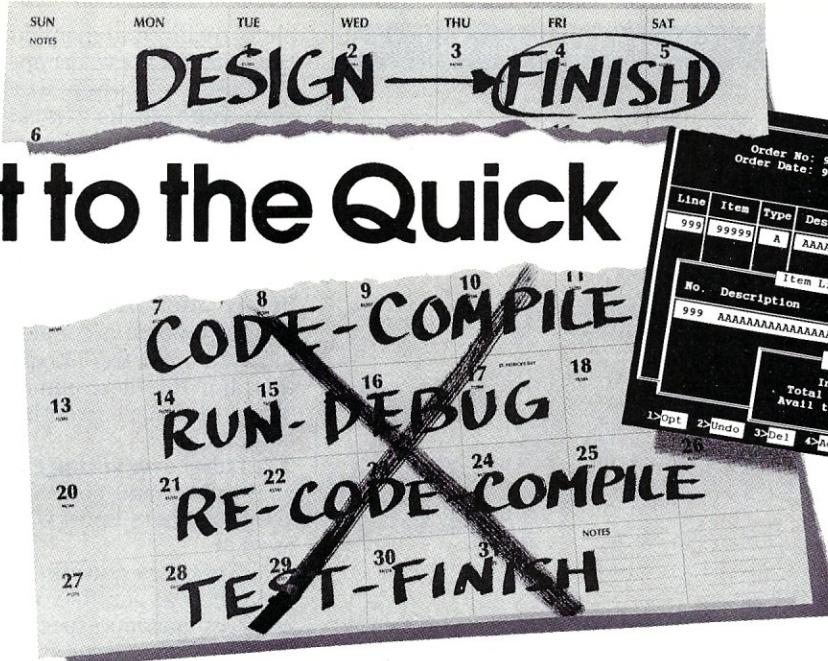
Invisible Software, Inc.
481 47th Ave.
San Francisco, CA 94121
(415) 221-0916

	Model 100	Model 200	Model 200/A
Compatibility	IBM PC/XT/AT	IBM PC/XT/AT	IBM PS/2 Micro Channel
Speed	0.7 Mbps	1.8 Mbps	1.8 Mbps
Price	\$199.00	\$299.00	\$399.00

For more information call (415) 221-0916 or write.

CALL ADVERTISER DIRECTLY

Attn
Btrieve®
programmers.



MAGIC PC ELIMINATES CODING . . . CUTS MONTHS OF DATABASE DEVELOPMENT!

Time is money. And coding a DBMS application like Accounting or Order Entry takes a lot of both. Simply because hacking out mountains of code with your RDBMS or 4GL is too slow. Not to mention the time to rewrite if you make a mistake or change the design.

EXECUTION TABLES ELIMINATE CODE!

Magic PC cuts months of your application development time because it eliminates coding. You program with the state-of-the-art Execution Tables in place of conventional programming.

HOW DOES IT WORK?

Magic PC turns your database design scheme directly into executable applications without any coding. Use Execution Tables to describe only what your programs do with compact design spec's, free from lengthy how to programming details. Each table entry is a powerful non-procedural design instruction which is executed at compiled-like speed by a runtime engine. Yet the tables can be modified "on the fly" without any maintenance. Develop full-featured multi-user turn-key systems with custom screens, windows, menus, reports and much more in days — not months! No more low-level programming, no time wasted . . .

MAGIC PC™

The *Visual* Database Language



"Magic PC's database engine delivers powerful applications in a fraction of the time . . . there is no competitive product."



"Overall, Magic PC is one of the most powerful DBMS packages available."

- Quick Application Generator
- BTRIEVE® — based multi-user RDBMS
- Visual design language eliminates coding
- Maintenance-free program modifications
- Easy-to-use Visual Query-By-Example
- Multi-file Zoom window look-ups
- Low-cost distribution Runtimes
- OEM versions available

ATTENTION BTRIEVE® USERS

Now you can quickly enhance your BTRIEVE®-based applications beyond the capabilities of XTRIEVE® and RTRIEVE®. Use Magic PC as a turn-key BTRIEVE® Application Generator to customize your applications without even changing your existing code.



19782 MacArthur Boulevard, Suite 305
Irvine, California 92715

TLX: 493-1184 FAX: 714-955-0199

DATABASE PROGRAMMERS

Join the thousands of professional database programmers and vertical market developers who switched to Magic PC from dBase®, R:BASE®, Paradox®, Clipper®, Dataflex®, Revelation®, Basic, C, Pascal, etc.

TRY BEFORE YOU PAY

We're so sure you'll love Magic PC — we'll let you try the complete package first. Only a limited quantity is available, so call us today to reserve your copy. Pay for Magic PC only after 30 days of working with it.* To cancel . . . don't call . . . simply return in 30 days for a \$19.95 restocking fee.

OR PAY NOW AT NO RISK

Pay when you order and we'll waive the \$19.95 restocking fee so you have absolutely no risk.

SPECIAL OFFER \$695

\$199



Magic LAN multi-user — \$399
Magic RUN — call for price

**Order Now Call:
800-345-MAGIC**
In CA 714-250-1718

MS

Add \$10 P&H, tax in CA. International orders add \$30.

*Secured with credit card or open P.O. Valid in US.

Dealers welcomed

CALL ADVERTISER DIRECTLY

by Patrick H. Corrigan

Twisted Pair

Networking Part Two: The Wiring Closet

For most of us, the telephone wiring closet is a place of mystery with strange connector blocks and a maze of wire going in all directions. With the increased use of unshielded, twisted-pair (UTP) telephone wire for local area networks, it is important that LAN designers, integrators, and installers understand how to work in the phone closet.

Phone Closet Etiquette

Since the breakup of AT&T, office building phone closets have become accessible to most vendors and installers. Different systems for different building tenants are now often cabled through the same phone closet. An improper installation for one building tenant, however, can often cause problems for others. Like other shared facilities, peaceful coexistence requires that certain rules of etiquette be followed by all concerned. Unfortunately, these rules of etiquette are frequently broken, with new and inexperienced and/or uncaring installers often being the worst offenders. Since most LAN installers currently fit the category of "new and inexperienced" with respect to phone closets and phone wiring, a review of some of the rules may be in order. This is by no means a complete list, and other rules also apply in specific situations.

1. Do not disrupt service for others. Accidentally disrupting phone service is not difficult to do. Be careful!

2. Do not cut existing lines. Lines

Patrick H. Corrigan is a partner in The Corrigan Group — Information Services, an independent consulting firm specializing in local area networks and office automation based in Corte Madera, California.

may have been installed by others for future expansion.

3. Get permission. Many offices have strict rules concerning building cabling. In addition, local building codes and licensing issues may be involved in any cabling. You may need the following: permission from the line owner(s) to use any existing lines; permission from the building management to do any cabling work in the building; building permit(s); and/or appropriate licenses. In most areas, you need a contractors license to actually install cable in a building. In addition, union contracts may require union personnel to perform any actual cable installation.

4. Use similar supplies. If you are expanding or modifying an existing cabling system, use connector blocks and components similar to those already in use. This makes it easier for others who may have to work on the same system.

5. Use correct connectors and adapters. Using incorrect connectors can result in poor line quality. This is annoying for voice communications and disastrous for data.

6. Follow accepted color codes. Premise wiring systems use a standard set of color codes for cables. It is important to learn them and use them. (This will be covered in the next *LANDSCAPE* column.)

7. Plan your use of closet space. In most phone closets space is limited. Plan for expansion, but leave space for others.

8. Use the correct cable. If you are installing cable, make sure it meets network system requirements. In addition, make sure any cable installed meets any requirements or specifications prescribed by building management and local building codes. For example, teflon-coated "plenum" ca-

ble is required in many localities for cable routed above dropped ceilings.

9. Take care when working with connector blocks. It is easy to "short out" lines when working with connector blocks.

10. Use the proper tools. You are less likely to damage equipment if you use the proper tools.

11. Learn to use unfamiliar tools before attempting an installation. You may waste a couple of connectors and a few feet of cable by doing this, but this is a minor cost compared to that of repairing damaged equipment.

12. Label everything clearly. Label your cables and connectors so that you and others know which lines go to what.

13. Neatness counts. Keep all wiring as neat as possible. Wire management components especially designed for the phone closet are available to make this easier.

The Right Tools

There are a few tools that are useful when installing UTP LANs. They are:

- **Modular connector crimper.** These range in price from about \$8 to \$125. If you are going to make your own modular cables and plan to install more than one system, invest in a good crimper. The more expensive ones usually last longer, work better, and can usually be adapted for different types of modular connectors. AMP makes a good one, and I am sure there are others. (AMP's part # 231666-7 crimps 2-, 4-, 6-, and 8-position connectors.)
- **Punch-down tool.** This is a tool used for attaching wires to S66 Quick-Connect Blocks found in most wiring closets. A good one will cost about \$40 to \$45. (Siemon part # S814-66 or MOD-TAP part # 26-505 are two of the better ones.)
- **Probe Pick.** A tool used for removing wires and bridging clips from S66 blocks. (Siemon part # PROBE PIC.)
- **Cable stripper.** This tool is designed for use with 2-, 3-, and 4-pair telco cable. It should cost less than \$5 and can save a lot of time. (Siemon part # S953A or AT&T part # 953A-50.)

Cable and Connectors

Most UTP LAN systems are designed to use D-Inside Wire (DIW). D-Inside Wire is a balanced, twisted-pair, 24-gauge copper, unshielded cable. Although it is made in several sizes, the most common sizes are 3-, 4-, and 25-pair. DIW is not as susceptible to

Table 1. Common DIW cable pinouts with 25-pair color codes

25 Pr Conn.#	50 Pin Conn.#	Wire Color (Wire/Stripe)	2-pair Pin #	3-pair Pin #	4-pair Pin #
1	26	White/Blue	4	4	5
	1	Blue/White	3 Group	3	4
2	27	White/Orange	2 1	2 Group	1
	2	Orange/White	5 _____	5 1	2 Group
3	28	White/Green	4	1	3 1
	3	Green/White	3 Group	6 _____	6
4	29	White/Brown	2 2	4	7
	4	Brown/White	5 _____	3	8 _____
5	30	White/Slate	4	2 Group	5
	5	Slate/White	3 Group	5 2	4
6	31	Red/Blue	2 3	1	1
	6	Blue/Red	5 _____	6 _____	2 Group
7	32	Red/Orange	4	4	3 2
	7	Orange/Red	3 Group	3	6
8	33	Red/Green	2 4	2 Group	7
	8	Green/Red	5 _____	5 3	8 _____
9	34	Red/Brown	4	1	5
	9	Brown/Red	3 Group	6 _____	4
10	35	Red/Slate	2 5	4	1
	10	Slate/Red	5 _____	3	2 Group
11	36	Black/Blue	4	2 Group	3 3
	11	Blue/Black	3 Group	5 4	6
12	37	Black/Orange	2 6	1	7
	12	Orange/Black	5 _____	6 _____	8 _____
13	38	Black/Green	4	4	5
	13	Green/Black	3 Group	3	4
14	39	Black/Brown	2 7	2 Group	1
	14	Brown/Black	5 _____	5 5	2 Group
15	40	Black/Slate	4	1	3 4
	15	Slate/Black	3 Group	6 _____	6
16	41	Yellow/Black	2 8	4	7
	16	Black/Yellow	5 _____	3	8 _____
17	42	Yellow/Orange	4	2 Group	5
	17	Orange/Yellow	3 Group	5 6	4
18	43	Yellow/Green	2 9	1	1
	18	Green/Yellow	5 _____	6 _____	2 Group
19	44	Yellow/Brown	4	4	3 5
	19	Brown/Yellow	3 Group	3	6
20	45	Yellow/Slate	2 10	2 Group	7
	20	Slate/Yellow	5 _____	5 7	8 _____
21	46	Violet/Blue	4	1	5
	21	Blue/Violet	3 Group	6 _____	4
22	47	Violet/Orange	2 11	4	1
	22	Orange/Violet	5 _____	3	2 Group
23	48	Violet/Green	4	2 Group	3 6
	23	Green/Violet	3 Group	5 8	6
24	49	Violet/Brown	2 12	1	7
	24	Brown/Violet	5 _____	6 _____	8 _____
25	50	Violet/Slate	-	-	-
	25	Slate/Violet	-	-	-

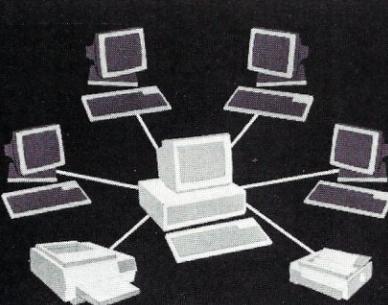
Note: 2 and 3 pair pinouts refer to RJ-11 type modular jacks and plugs. 4 pair pinouts refer to RJ-45 type jacks and plugs.

Discover The Multi-User Mini That's Hiding In Your 386 PC

Now you can have a complete DOS-based multi-user system with up to 11 people sharing a standard 386 PC—without the cost and complexity of a LAN.

Using inexpensive terminals and the awesome power of the 80386 CPU, Music/386 lets each user on the system run any program and access any peripheral or data file, simply and inexpensively.

And best of all, Music/386 is fully-compatible with standard PC-DOS, so you can continue using all your favorite applications.



Multi-User Computing Has Never Been This Easy... Or This Compatible

Only Music/386 gives your PC so much power at such low cost:

- Install and start multi-terminal computing in less than 8 minutes
- Maintain the highest compatibility with DOS, because you still have real PC-DOS
- Share databases, programs, printers and peripherals
- Perform multi-tasking (as many as 255 different sessions on the main console)

Call today to find out how easily and inexpensively you can move up to multi-user computing.

Music/386

Multi-User System for Interactive Computing
BRISTOL INFORMATION SYSTEMS

84 North Main Street
Fall River, MA 02722-1031
(508) 679-1051

CIRCLE 117 ON READER SERVICE CARD.

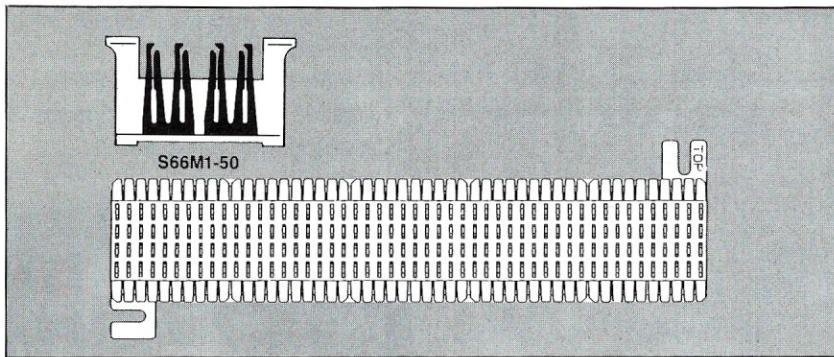
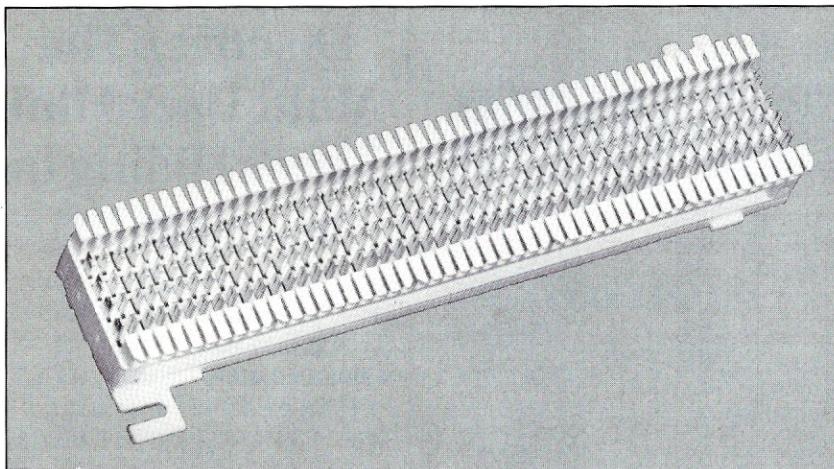


Figure 1. S66-type quick connect blocks

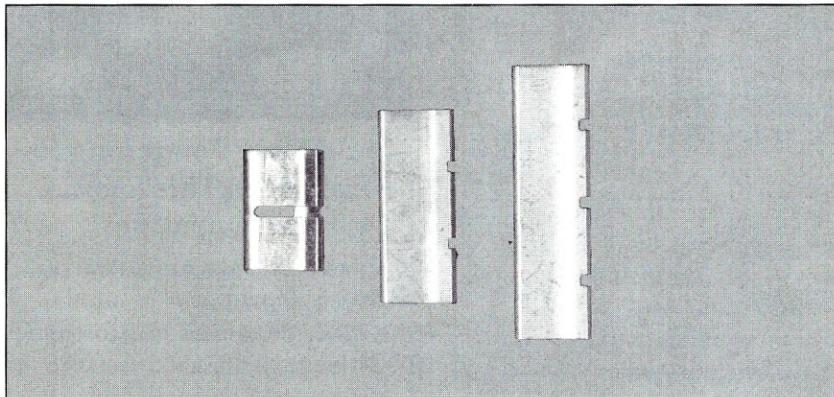


Figure 2. Bridging clips

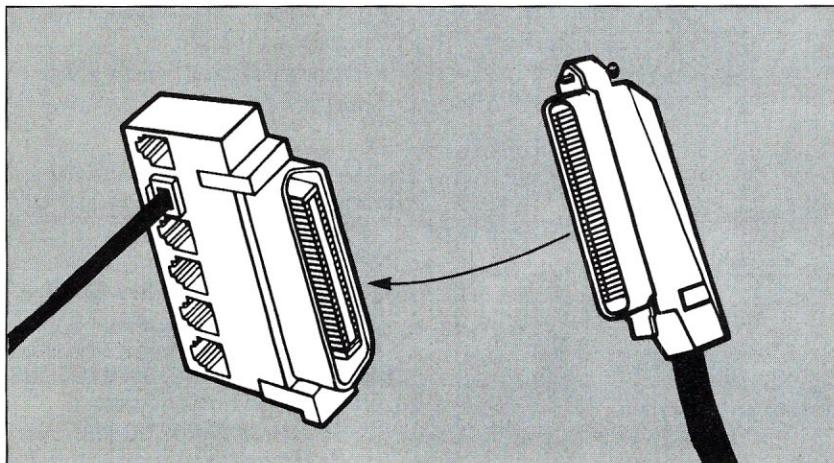


Figure 3. A 50-pin to six 8-pin adapter

noise and crosstalk as other unshielded cables. IBM Type 3 cable is D-Inside Wire. DIW is easy to identify. It has a grey or beige vinyl jacket, and each pair has a characteristic color code. The first four pairs are colored as follows:

- Pair 1: White with blue band, blue with white band
- Pair 2: White with orange band, orange with white band
- Pair 3: White with green band, green with white band
- Pair 4: White with brown band, brown with white band

(See Table 1 for a 25-pair color code listing.)

Almost all connections in the wiring closet are made through S66-type quick connect blocks.

Three types of connectors are commonly used when connecting UTP LANs: six-position modular connectors, usually referred to as RJ-11 connectors; eight-position modular connectors, usually referred to as RJ-45; and 50-pin connectors for 25-pair cable.

Almost all connections in the wiring closet are made through connecting devices called "S66-type Quick-Connect Blocks" or "punch-down blocks" (Figure 1). Although available in several configurations, these usually have two double rows of 50 "punch-down" connectors. With the use of a special punch-down tool, solid copper DIW wires can be quickly and easily connected without stripping off the insulation. These blocks can be purchased with 50-pin connectors pre-installed for easy attachment of 25-pair cables.

S66 blocks are usually attached to a backboard or distribution frame using special stand-off brackets. Cables can be routed behind the blocks, which snap onto the stand-off brackets.

Cables should be attached to S66

blocks in a standard fashion. With 25-pair cable, for example, pair #1 should be at the top and pair #25 should be at the bottom. Two-, three-, and four-pair cables are usually cabled in groups, with the first group starting at the top and the last group at the bottom of the block (see Table 1).

Connections between circuits can be made by cabling two circuits together on the same side of the S66 block, or by cabling circuits to either side of the block, and then using jumper wires or bridging clips to cross-connect the two circuits (Figure 2).

Modular connections can easily be made to S66 blocks by using adapters or patch panels, which usually attach to the block with a 50-pin connector (Figure 3). In addition, several companies supply blocks with adapters pre-installed (Figure 4).

Modular connector cables should use twisted pairs instead of parallel wires (most flat phone cables are parallel). For Synoptics UTP LattisNet (Ethernet over unshielded twisted-pair) and StarLAN, AT&T D8W or DW8A-DE cables or equivalent are usually recommended. □

Did you find this article particularly useful?
Circle number 9 on the reader service card.

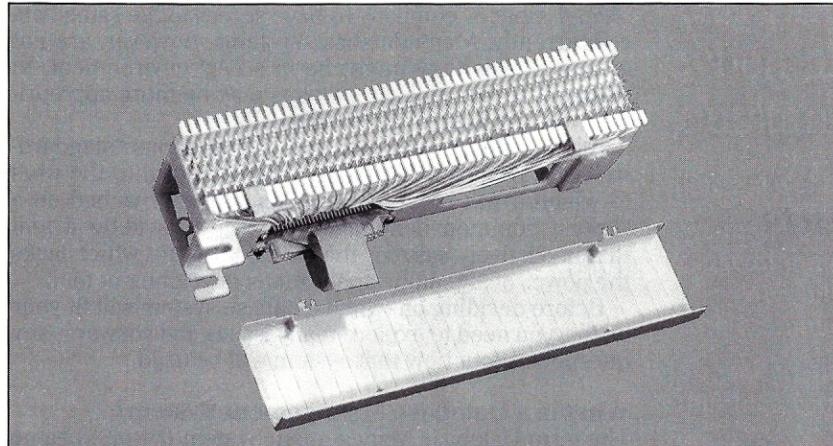


Figure 4. S66 block with pre-installed adapter

**No
Matter
Where
You Go,
There
We Are!**

*For change of address,
Affix present mailing label here*

New Address _____
City _____ State _____ Zip _____

Mail entire ad to:

Micro/Systems Journal, P.O. Box 3713, Escondido, CA 92025

Product Information

AT&T Network Systems
111 Madison Avenue
Morristown, NJ 07960
(201) 631-7500
Circle reader service #269.

AMP Products Corporation
Valley Forge, PA 19842
(215) 647-1000
Circle reader service #270.

MOD-TAP System
P.O. Box 706
Harvard, MA 01451-9987
(617) 456-3500
Circle reader service #271.

The Siemon Company
76 Westbury Park
Watertown, CT 06795-0400
(203) 274-2523
Circle reader service #272.

SynOptics Communications, Inc.
329 North Bernardo
Mountain View, CA 94043
(415) 960-1100
Circle reader service #273.

Introducing

NANODISK

"Disk Cache for the IBM PC"

Make your floppy drive and hard disk run close to RAM disk speeds. Dramatic speed improvement for most programs. Supports cache of any size in main or expanded memory.

Requires IBM PC/XT/AT or true clone.

only **\$29.95**

MultiDos Plus

"multitasking for the IBM-PC."

Ideal for developing applications in process control, data acquisition, communications, and other areas. Check these features which make **MultiDos Plus** an unbeatable value.

- Run up to 32 DOS programs concurrently.
- Operator commands to load/run programs, change priority, check program status, abort/suspend/resume programs.
- Programmatic interface via INT 15H for the following.
 - * Intertask message communication.
 - * Task control by means of semaphores.
 - * 256 priority levels.
 - * Suspend task for specified interval.
 - * Spawn and terminate external and internal tasks.
 - * Disable/enable multitasking.
 - * and more!

Requires IBM PC/XT/AT or true clone, and enough memory to hold **MultiDos Plus** (48 KB) and all your application programs.

\$24.95 or \$99.95

With source code
(Written in Lattice C
and Microsoft Assembler.)

Outside USA add \$5.00 shipping and handling.
Visa and Mastercard orders only call toll-free: 1-800-872-4566, ext. 350, or send check or money order (Drawn on U.S. Bank Only) to:

NANOSOFT

13 Westfield Rd, Natick, MA 01760
MA orders add 5% sales tax.

CALL ADVERTISER DIRECTLY

What To Look for In a LAN Database

by Patrick H. Corrigan

*Network databases
have to be
chosen according
to data requirements
and user
sophistication —
one size doesn't fit all.*

Local area networks have extended the usefulness of database software by allowing a whole department or company to have access to the same data concurrently. Many database systems, however, are not as effective as they might be in a LAN environment. In addition, some database systems may be more appropriate for a given requirement than others.

All too often, companies and organizations "standardize" on a particular database without regard to the uses to which it will be put. This is usually approached on a "lowest common denominator" basis, providing a tool that is relatively easy for the casual user, but which lacks the power of a genuine applications development tool.

Before deciding on which database system will fit your needs, you need to explore many issues and answer many questions about how that system will be used.

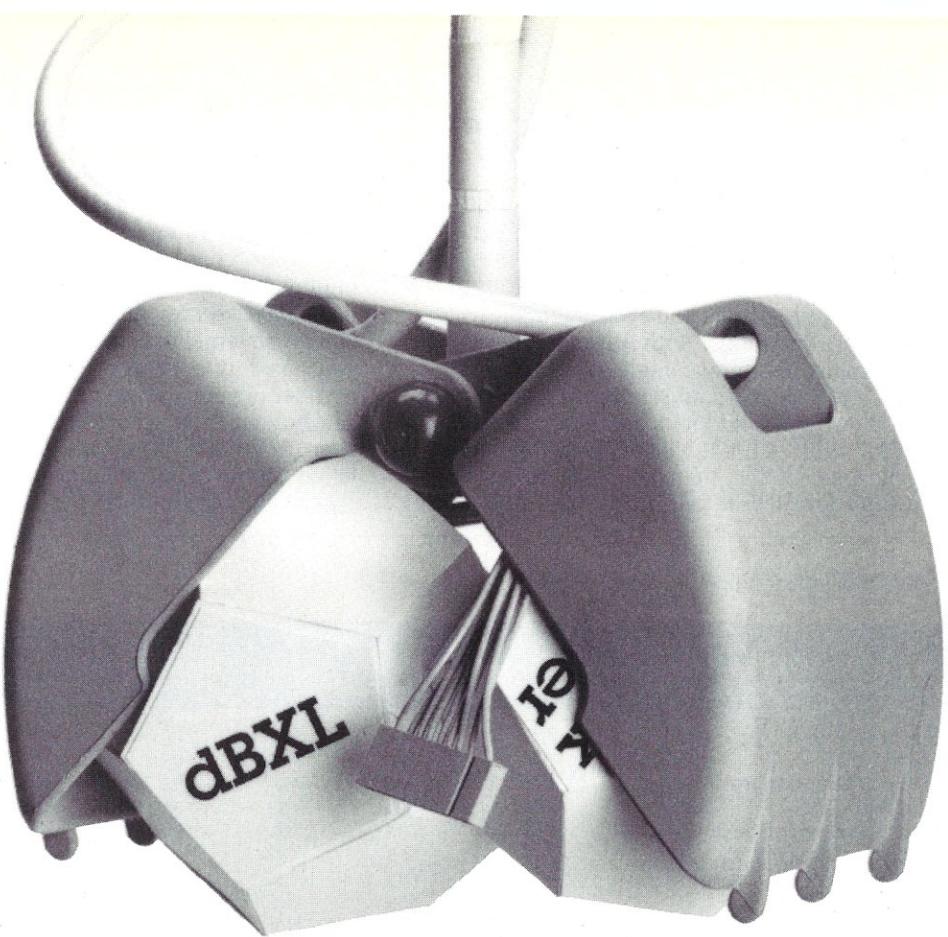
What is a Database Management System?

The term Database Management System (DBMS) can be applied to a wide range of software products, from simple filing systems to complete applications development environments. Basically, a DBMS is composed of two parts: a file management system; and a user and/or programmer interface to the filing system, often including complete application development tools.

The functions of both parts of the DBMS are important, and one or both may help determine the suitability of a DBMS for a particular situation or need.

Not too many years ago, all DBMSes ran on large mainframe computers, with armies of programmers on tap to maintain them. In that environment, the needs were usually thoroughly assessed before any software was purchased or any code was written. In today's PC

Patrick H. Corrigan is a partner in The Corrigan Group — Information Services, an independent consulting firm specializing in local area networks and office automation based in Corte Madera, California.



environment, however, the users who previously were only allowed to enter and retrieve data on the mainframe are often the database programmers. Database software is often purchased with little thought to its appropriateness for a given purpose. This buying approach, while perhaps workable on a stand-alone PC, can be a problem in the LAN environment where different levels of users may need access to the same data. A DBMS designed to be used by an occasional user for simple list management would probably be completely unusable for someone trying to create a complex inventory and invoicing application. In addition, a DBMS optimized for freeform query of data stored in tabular form may not be the approach best for "turnkeyed" data entry and transaction processing.

To help determine what DBMS is appropriate, you need to ask some questions:

1. *Who is designing the database files and applications?* If all or many users in an organization are creating databases and applications for their own use, you may want to place emphasis on design tools that require little, if any, programming. If a group of database programmers or configurators are developing applications to be used by others for data entry and retrieval, flexibility and programmability may be primary concerns. Although many products claim to fill both needs, few, if any, really do.
2. *Who is using the database?* Are the users knowledgeable enough about PCs and software so they can handle queries and *ad hoc* reports, or are they primarily "data entry clerks" who need well-defined, easy-to-use, preprogrammed applications?
3. *What is the orientation of the DBMS?* Some DBMS systems are oriented toward the data structure, while others are oriented toward the user interface. For example, the table orientation of Paradox is very apparent to the user (unless you have an

extremely good programmer!) while the data structure is transparent to a user of a DataFlex application. The table orientation of Paradox is usually better in situations where data must often be retrieved in an *ad hoc*, non-structured way, while the "screen" orientation of DataFlex is often better for "turnkey" applications.

4. *How is data retrieved?* Sorting is fine for small files, but indexed retrieval and reporting is almost a must for efficient use of large databases. Although many database systems provide for indexing, they don't all do it the same way. Some allow multifield indexes; some update indexes immediately on-line, some allow partial index look up.
5. *Can data be entered into multiple files (tables) through the same screen?* While most sophisticated DBMS systems allow data to be entered into multiple files through the same screen, many require that each file have its own separate data window.
6. *How much control does the applications designer have over the appearance of screens and output?* Can the programmer completely control the screen design? Is this control available through the easy applications generation tools? Can the programmer design custom report formats?
7. *Are users locked out of records in use by others, or is there some method to provide concurrent access?* This is not important in all environments, but it can be critical in particular situations.
8. *Can reporting and updating be performed concurrently by different users?* If the database has to be "locked" to run a report, other users will be effectively locked out, and valuable time may be lost.
9. *Is there access to and from other applications and files?* Does the database have file import and export capabilities? Are there interfaces for programming languages?

10. *Are applications and files easily transportable?* Is it difficult to move a specific set of files and/or applications from one database to another?
11. *What does it really cost?* The cost per user on a LAN database varies greatly from product to product. Vendors have different pricing and licensing schemes. For example, some systems are priced by user and some are priced by file server, and even within these schemes there are variations. With Advanced Revelation, for example, you buy a single-user package at a retail price of \$950, then add users with "bump disks," that allow you to add four users for a retail price of \$495. With R:BASE you initially have a three-user license (\$725), then you may add six users at a time for \$695 retail, or you can buy a

single-server license with an unlimited number of users for \$2,695.

Many DBMSes provide runtime systems that allow you to distribute applications for a nominal fee. Others allow you to distribute compiled code for a low or no charge. The costs vary. The DataFlex multiuser runtime for DOS LANs retails at \$300 dollars per server and provides the full DataFlex Query system. For \$9.95 (no, that's not a misprint!) you can buy a license to distribute an unlimited number of your Paradox applications. For \$200 you can buy an Advanced Revelation runtime for one user. Additional users are added using the Revelation "bump disks" at \$495 for four users.

12. *What about performance?* Performance and response

Table 1. A Comparative Look at LAN Databases

LAN DATABASES	DataPerfect	DataFlex	Paradox	R:BASE	Advanced Revelation	dDBXL/Quicksilver
Users locked out of records in use?	No	No	No	No	No ¹	Yes
Easy development tools?	Yes	Yes	Yes	Yes	Yes	Yes ²
Consistent with procedural language?	N/A ³	Yes	Yes ⁴	Yes	No	Yes
On-line help for developers?	Yes	No	Yes	Yes	Yes	Yes
Configurable on-line application help?	Yes	Yes	No	Yes	Yes	Yes
Multiple on-line indexes available?	Yes	Yes	Yes	Yes	Yes	Yes
Indexes updated on-line?	Yes	Yes	Yes	Yes	No	Yes
Multifield indexes available?	Yes	Yes	No ⁵	No	No	Yes
Display/update multiple files per screen window?	No	Yes	No ⁶	No	Yes	Yes
Developer has complete screen control?	No	Yes	Yes	Yes	Yes	Yes
Easy development tools provide this control?	No	Yes	No	Yes	No	No
Compiler available?	N/A ⁷	Yes ⁸	No	Yes	Yes ⁹	Yes
Can global updates be performed?	Yes ¹⁰	Yes ¹¹	Yes	Yes ¹²	Yes	Yes
Report and update concurrently?	Yes	Yes	Yes	Yes	Yes	No
Import & export files	Yes	Yes	Yes	Yes	Yes	Yes
Database accessible to programming languages?	No	Yes ¹³	Yes ¹⁴	Yes ¹⁵	No	Yes
Transport individual applications to other systems easily?	Yes	Yes	Yes	No	No	Yes
Are runtime systems available?	No ¹⁶	Yes	Yes ¹⁷	Yes	Yes	No
If database is not purchased, are runtime systems required for applications?	Yes	Yes	Yes	Yes	Yes	Yes
Is record locking automatic?	Yes	Yes	Yes	Yes	Yes	No
Can automatic locking be overridden?	No	Yes	Yes	Yes	Yes	N/A
Performance affected by available RAM?	No	No	Yes	?	?	?
Performance affected by database size?	No	No	Yes	?	?	?
Can applications interoperate over multiple operating systems?	No	Yes	Yes	?	No	No

Notes:

1. Although a record may be locked, other users may view it.
2. DBXL, WordTech's dBASE III look-alike, provides INTRO, a menued front-end for novice users. While not a true applications generator, INTRO will allow a new user to create simple mail list programs.
3. DataPerfect does not have a procedural language.
4. When building an application from the Paradox menu, the developer may create a "script" file that can then be enhanced with Paradox Application Language (PAL).
5. Primary Key may consist of first several fields in database table.
6. Multiple files in a window may only be updated in a batch mode.
7. Because DataPerfect does not have a procedural language, compiling is not necessary to improve performance.
8. DataFlex applications are always compiled.
9. Programs created with R:BASIC may be compiled.
10. Global updates are handled through reporting system.
11. Global updates require programming.
12. Global updates can be performed within a particular table.
13. Programming languages are accessible through the DataFlex "C Source and Library."
14. 'C' routines are available.
15. Program interface for MicroSoft languages is available as separate product.
16. Runtime system is currently in development.
17. Unlimited run-time license costs \$9.95.

(data retrieval, saves/updates, and reports) of LAN databases can vary from relatively fast, as with DataFlex, to relatively slow, as with Advanced Revelation. In general, several factors can affect performance of all DBMSes, including LAN speed, workstation processor type and speed, file server type and speed, etc. With some DBMSes, performance can be affected by available workstation RAM and/or the size of the database. DBMSes that use sort routines will generally slow down as the database gets larger, while DBMSes that use indexed retrieval will generally offer fairly consistent performance, regardless of the database size.

13. **What about data security?** Depending on your particular LAN environment, a DBMS that has its own security system may be important, or the security may just be redundant and get in the way. If security is important, make sure that the implementation works in your specific environment. Different vendors have different ideas of what security should be, and those ideas may not necessarily coincide with yours.
14. **What operating environments are supported?** With the advent of OS/2 and the increasing interest in UNIX and Xenix, this question becomes more important. If you are planning to move to another operating environment, it would be nice to move your DBMS —lock, stock, and data file—with you. If the vendor supports multiple environments, you might want to find out if the applications that you have written are transportable.
15. **Can applications interoperate over multiple operating systems?** This is a question that wouldn't have been asked much a few years ago, but LANs are increasingly connecting systems that use different operating systems. DataAccess sources say that the same DataFlex files can be concurrently accessed by LAN users using DOS, OS/2, UNIX, Xenix, AIX, and VAX VMS. In addition, they say applications written in DataFlex can be transported within the same processor family (8086, 286, 386, and the VAX are considered one family; 68000-based systems are another) without recompiling. Fox Software, the publishers of FoxBASE (not profiled here), indicate

that Macintosh users will be able to concurrently share database files with PC users on the same network using the upcoming Macintosh version of Novell NetWare.

16. **What about support?** Support plans vary as much as DBMSes. Some examples: Users of DataPerfect receive unlimited installation and configuration assistance (no design support, however) and a toll-free number to call. R:BASE users receive 30 days of free support, after which they can buy a one-year contract that can cost \$175 to \$600 for a single user, depending on the products being supported. DataFlex users receive unlimited support on a toll line, a monthly newsletter, and monthly bulletins on known bugs(!), patches, and workarounds. Many vendors offer different kinds of extended support programs for groups such as software developers, major accounts, etc.

Profiles of Database Products

Now that we have listed the points to consider, let's examine some of the popular database packages currently available that are suitable for network applications.

DataPerfect

DataPerfect from WordPerfect Corporation is a DBMS/Application Generator designed to eliminate the need for programming. Using function keys in a manner similar to WordPerfect, it allows the user to create applications by drawing screens and defining fields. Among other features, DataPerfect provides up to 80 related files per database, multiple on-line indexes, multifield indexes, and WordPerfect Corporation offers toll-free support. DataPerfect's preprogrammed facilities should make it possible for a novice user to create effective applications relatively quickly. Because of the lack of a procedural language, DataPerfect does not offer all of the flexibility that a programmer might desire. DataPerfect is also extremely easy to install.

DataFlex

DataFlex from DataAccess Corporation has been on the market since 1981. Originally written to run under

DataPerfect

Import/export file format: ASCII, WordPerfect, WordPerfect Merge

Operating system(s) supported: DOS

LAN operating system(s) supported: NetWare, IBM PC LAN Program, 3+, MS-Net

Product licensing: Per user

Standard support program: Unlimited toll-free telephone for installation and troubleshooting

Additional features: Record changes are displayed on all screens with that record open. Stand-alone and LAN versions are the same. No special LAN installation required.

Costs:

First user	\$595
Each additional user.	\$150

Contact:

WordPerfect Corporation
1555 North Technology Way
Orem, UT 84057
(801) 227-4020

Circle reader service #251

DataFlex

Import/export file format: ASCII, dBASE, others

Operating system(s) supported: DOS, VMS, AIX, UNIX, Xenix, OS/2

LAN operating system(s) supported: NetWare, IBM PC LAN Program, 3+, LifeNet, others

Product licensing: Per server

Standard support program: Unlimited toll-free telephone support for installation and troubleshooting. A special developer program is available. Additional services are available.

Additional features: Datafiles are transportable among all supported operating systems. Compiled applications are transportable within a given processor family. Full graphics support.

Costs:

Development license per file server	\$1,250
-------------------------------------	---------

Contact:

DataAccess Corporation
1400 S.W. 119th Avenue
Miami, FL 33186
(305) 238-0012

Circle reader service #252

TurboDOS (a multiuser operating system that runs CP/M programs), DataFlex has always been a multiuser DBMS. DataFlex provides a straightforward, applications development environment. Although primarily geared to programmers, DataFlex provides facilities such as AutoDef and Query that allow a novice user to create working applications and reports. Code generated with AutoDef and Query is completely consistent with DataFlex's very complete procedural language, which provides extensive screen manipulation capabilities, extensive math functions, and full graphics support.

DataFlex has long been recognized as one of the fastest LAN DBMSes available, both in terms of data retrieval and reporting, and in terms of application development time. Because of its extensive use of multi-field indexes for retrieval and reporting, and its built-in menu system,

DataFlex is well-suited for creating "turnkey" applications that require the user to have little or no knowledge of the DBMS itself. In addition, because DataFlex currently runs in so many operating environments (DOS, OS/2, UNIX, Xenix, AIX, and VAX VMS) applications are very transportable. DataFlex provides an extensive, easy-to-use, multi-file query function that can provide *ad hoc* reports as well as create editable source code for more formal reports.

Paradox

Paradox is a DBMS/Applications generator that uses a Lotus-like menuing system. This is probably very appropriate, since Paradox organizes data into "tables" that look to the casual user like Lotus spreadsheets.

Paradox allows users to build databases and applications without programming, but by simply selecting items from the Lotus-style menu. With its extensive help system, Paradox users should rarely have to refer to the documentation.

The easiest form of programming Paradox is to create a "Script" file. This is a file that repeats keystrokes used to perform a specific function. Script files can be modified using the Paradox Application Language (PAL) to add additional capabilities. Through the use of PAL, complex multi-file applications can be built with Paradox. Applications written with Paradox may be distributed in unlimited numbers if you buy a Paradox runtime license for \$9.95. In addition, individual Paradox applications are easily transportable.

Because of its structure and command set, Paradox is well suited to freeform queries. Paradox's indexing system leaves something to be desired, however. Because Paradox does not provide multi-field indexes (other than the primary key, in a limited fashion), retrieval of data based on multiple fields requires a selection process that can be fairly slow. As the database grows, this selection process can take longer, depending on the location of the desired record in the database.

R:BASE

R:BASE from Microrim is a distant cousin to a DBMS that Boeing Computer Services wrote for NASA called RIM (Relational Information Manager). Like Paradox, R:BASE organizes data in a database into tables. Unlike Paradox, which allows unlimited tables per database, R:BASE allows up to 80 tables with a maximum total of 800 columns (fields). The current release of R:BASE includes a subset

R:BASE

Import/export file format: A variety of types using the R:BASE File Gateway

Operating system(s) supported: DOS, OS/2

LAN operating system(s) supported: NetWare, IBM PC LAN Program, 3+, others

Product licensing: Per user or server. R:BASE can be licensed for a particular number of users or an unlimited single-server license may be purchased.

Standard support program: 30-day telephone support. Extended programs are available.

Additional features: Single-user applications can be made multiuser without change. Concurrency Control system displays other users' changes to a record.

Costs:

1 to 6 users	\$995
Unlimited use	\$2,695

Add-ons:

Clout	\$295
ERW	\$295
Program Interface	\$595
R:Tools	\$99
DB Graphics	\$295

Contact:

Microrim
3925 159th Avenue N.E.
Redmond, WA 98052-9722
(206) 885-2000

Circle reader service #253

Advanced Revelation

Import/export file format: ASCII, 1-2-3, dBASE

Operating system(s) supported: DOS

LAN operating system(s) supported: NetWare, IBM PC LAN Program, 3+, others

Product licensing: Per user

Standard support program: 90-day telephone support. Extended programs are available.

Additional features: Records can be of variable length with multi-value fields.

Costs:

One user	\$950
Four additional users	\$495

Contact:

Revelation Technologies
3633 136th Place S.E., Suite 106
Bellevue, WA 98006
(206) 643-9898

Circle reader service #254

Paradox

Import/export file format: ASCII, 1-2-3, PFS, dBASE, Symphony

Operating system(s) supported: DOS, OS/2

LAN operating system(s) supported: NetWare, IBM PC LAN Program, 3+, others

Product licensing: Per user

Standard support program: Unlimited telephone support for installation and troubleshooting.

Additional features: Record changes are displayed on all screens with that record open.

Costs:

Per user	\$725
For 6 users	\$995

Contact:

Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066
(408) 438-8400

Circle reader service #255

of the SQL (Structured Query Language, usually pronounced "Sequel") command set, in addition to the R:BASE command set.

R:BASE provides extensive facilities to make things easier for the novice user, including Prompt By Example (PBE) for R:BASE commands, and the application development tools Definition EXPRESS, Forms EXPRESS, Reports EXPRESS, and Application EXPRESS. Microrim also supplies CLOUT, a "natural language interface" to R:BASE, as a separate add-on product. Other products available separately include the R:BASE Program Interface, to allow MicroSoft languages to access the R:BASE database; Extended Report Writer, a report writer with greater capabilities than the one supplied with the product; and R:BASE RUNTIME, a system for distributing R:BASE applications.

The R:BASE support plan is one of the most expensive in the industry. Microrim provides 30 days of free support, after which you must buy a support plan. The individual plan costs \$175 per registered user for R:BASE, plus \$85 for each R:BASE companion product you are using (DB Graphics, Clout, Extended Report Writer, Program Interface, and R:Tools). This means that support for one user

The table orientation of Paradox is usually better in situations where data must be retrieved in a non-structured way. DataFlex is better for "turnkey" applications.

can cost up to \$600 annually, while support costs for eight users could be between \$1400 and \$4800 annually. Microrim also offers corporate support plans that can reduce costs for a large number of users.

Advanced Revelation

In some ways, Advanced Revelation from Revelation Technologies is the most powerful applications development tool available for PC LANs. It provides variable-length records and fields, multi-value fields (e.g., a telephone number field in a single record that could contain any quantity of telephone numbers), database re-definition "on the fly," and much more. The "Paint" function of Advanced Revelation even allows the novice user to create applications quickly and easily.

There is a price to be paid for all this power, however. In terms of data retrieval and updating, Advanced Revelation is one of the slowest DBMSes available. And, despite its slick Paint function, Advanced Revelation is almost totally oriented toward the experienced programmer. To go beyond the capabilities of Paint, you must learn to use the command driven "RLIST" report language and the "RBASIC" procedural language. It is difficult, if not impossible, to build on Paint-generated applications using RBASIC, which means the novice user must start from scratch after learning to use Paint.

Although individual applications developed in Advanced Revelation can be moved from one database to another, this requires preplanning in the early development stages — you need to know early on that you are going to move the application.

A minor annoyance is that error messages that can be generated through the Paint program are not documented in the manuals. If you know how to use RLIST, however, there is an error message file that is accessible, although this fact is not well-documented.

dBXL/Quicksilver/Networker Plus

dBXL from WordTech Systems is a self-proclaimed "dBASE III Plus SuperClone." Along with its companion products Quicksilver, a dBASE III and dBXL compiler, and Networker Plus, a LAN utility package, dBXL provides a complete, extended, dBASE-like LAN database system at considerably lower cost than dBASE III Plus. Quicksilver-compiled applications generally execute faster than interpreted applications. And Networker Plus provides many of the functions of the dBASE III Plus networking component, Administrator. In addition to record- and file-locking functions, Networker Plus provides some extras. These include two-way station-to-station communications that allow users to communicate interactively; screen transmission that allows one user to send a copy of a screen to another user, or capture another user's screen; and distributed batch processing, that allow workstations to off-load batch-oriented tasks to other stations.

Conclusion

In our culture, we tend to look for the one "best" product in any given category, ignoring the possibility that different products may fit different needs. Indeed, because so many types of products are grouped under the heading "DBMS," we are often comparing vastly dissimilar systems (see Table 1). In the area of LAN databases, although we all have our favorite(s), we may have to admit that "one size" does not necessarily "fit all." □

Did you find this article particularly useful?
Circle number 5 on the reader service card.

dBXL/Quicksilver

Import/export file format: ASCII, SILK, 1-2-3, PCX
Operating system(s) supported: DOS

LAN operating system(s) supported: NetWare, IBM PC LAN Program, 3+, others

Product licensing: Per server

Standard support program: One hour of free support. Other programs are available.

Additional features: Networker Plus provides for messaging and batch processing across a LAN.

Costs:

dBXL-LAN (including Networker Plus) \$599

Quicksilver (\$698 including Networker Plus) \$599

Networker Plus \$259

Note: The price of dBXL-LAN is scheduled to increase to \$799 October 15.

Contact:

WordTech Systems

P.O. Box 1747

Orinda, CA 94563

(415) 254-0900

Circle reader service #256

An Introduction to SQL

by Edward A. Dowgiallo

This primer explores the ins and outs of Structured Query Language, the most powerful database approach to be ported from the mini to the PC.

One of the hottest topics in databases is the emergence of SQL (Structured Query Language) on the PC. Support for SQL on the PC is coming from two directions as companies porting SQL implementations from minicomputers vie with traditional providers of PC DBMS software to gain control of this fledgling market. What does all this activity mean for developers? What type of development environment does SQL provide? Will it speed up the application development process? What does the language look like? We will explore some of the answers to these questions in this article.

Background

SQL is a nonprocedural language designed for the creation, maintenance, and manipulation of relational databases. The language has a firm theoretical foundation in mathematical set theory, which shows itself in the syntax of the language. SQL statements allow the easy manipulation of data in large groups as well as by individual elements. The chief problem for the user is developing the skill to formulate these statements.

As is the case with most technologies, SQL has its own jargon to help the user adapt to the mindset of the set model on which it is based. Several of these terms are used extensively in the literature, and a brief explanation of terminology is in order. A *file* is thought of as a two-dimensional array formed of rows and columns. Files are referred to as *tables* to enforce this idea. The rows consist of individual physical records and are referred to as either *rows* or, more formally, as *tuples*. The columns of a table consist of the individual fields within a record and are referred to as *columns*.

A Brief History of SQL

SQL originated as a project at IBM's San Jose Research Laboratory following the publication of a series of seminal papers introducing the concept of relational database

Ed Dowgiallo is an independent computer consultant based in Delran, New Jersey. In addition to writing for computer magazines, he does custom design and programming in the areas of databases, data communications, and operating system extensions.

management by E.F. Codd in 1969 and 1970. Publications on SQUARE and SEQUEL as interface languages for a relational database called System R appeared in the mid-1970s. SEQUEL evolved into SQL as work progressed in the late 1970s on IBM's first commercial relational database, DB2. This was naturally an IBM mainframe implementation, but it was quickly followed by an implementation on DEC minicomputers by Oracle Corporation and Relational Technologies. Since then, SQL has been implemented on a large number of different computers as a relational database interface, and in the last couple of years it has begun to migrate onto the high-end microcomputers.

In February 1987, the American National Standards Institute (ANSI) made SQL an official standard, which has been acknowledged by most of the companies implementing products based on it. This is an area where the buyer must beware since not all implementors have adopted the standard, especially among the existing microcomputer databases that are migrating toward SQL and where partial implementations abound. A month later, in March 1987, IBM adopted SQL as the database language standard for its System Application Architecture (SAA).

Initial acceptance of SQL by the mainframe user community has been slow, with many DP shops remaining committed to the investment they have already made in IDMS and IMS/VS. After 10 years, DB2 has only a small percentage of this installed base. SQL acceptance will increase gradually in the mainframe world and will largely hinge on the success of IBM selling its SAA approach.

What Does the Technology Offer?

The greatest advantage to users in an SQL environment is the *ad hoc* query and reporting capability. From the beginning, an interactive SQL user can perform fairly sophisticated data analysis that spans information in multiple tables. The ability to reduce data via group and set operations is provided, including simple statistical calculations. The ability to generate columnar reports that span multiple tables, with multiple sort levels, on almost arbitrary slices of the database is inherent to the structure of SQL. Admittedly, the ability to use this power will depend on the cleverness of the user and will come easiest to the mathematically inclined. Still, the ability to perform many analyses that used to require the aid of the MIS department will now be placed directly in the hands of users.

Training is another area where SQL offers advantages. Super users will tend to quickly feel at home with the set manipulation paradigm on which the technology is based. The training investment is protected by the fact that SQL is already fairly well standardized. Most implementations use a syntax identical to that implemented in DB2. The primary differences are in command options that provide additional abilities rather than variations in syntax. This similarity allows users to move easily from one vendor's implementation to another, even across machine environments. A person who has been trained to use SQL on a personal computer has also been trained to use it on a super-micro, a minicomputer, or even the company's mainframe. This similarity has even been formalized as an ANSI standard for SQL.

In fact, SQL implementations are becoming one of the major arenas for work on connecting computers with dissimilar architectures from multiple manufacturers into integrated networks, thus encouraging large corporations to embrace the technology as a unifying influence in their computing environments. The mechanism for accomplishing this has been support for multiple hardware platforms at the SQL vendor level and the provision of networking support as an integrated part of these prod-

ucts. Wide-ranging connectivity is really a side benefit. With the more sophisticated implementations, an interactive SQL user can actually perform multiple table operations, where the tables involved physically reside on different computer systems!

Many of the vendors bringing SQL to the PC are porting implementations originally written for the mainframe and minicomputer worlds. There are some drawbacks to this in terms of what users may be accustomed to in the way of user-interface technology, but this situation is being corrected through competitive pressure from the traditional PC database vendors that are also moving into SQL. More to the ultimate advantage of users is the fact that vendors with a mainframe or minicomputer heritage are already involved in application generator and CASE technologies. Thus, as their implementations are ported to PCs, these programmer productivity technologies are also being ported.

Another consideration involving programmer productivity and training is IBM's inclusion of SQL in its set of SAA standards. The net effect of this inclusion will be to provide the same degree of standardization and portability at the level of third-generation languages (3GLs) that interface directly to SQL engines as has been provided by ANSI at the interactive SQL level. Some vendors have already implemented support for this.

As part of the SAA standard set, SQL is playing a large role in the development of OS/2, especially the Extended Edition. Just about all the vendors in the OS/2 database arena have announced their intention to develop SQL-based products. These include database servers for network applications as well as powerful single-user versions of relational databases.

Finally, users can look forward to a whole range of add-on products and "bridge" products that provide a connection between SQL and already existing popular packages. A good example of this is Oracle for 1-2-3, which is already on the market. This product allows Lotus 1-2-3 users to use SQL interactively from within a 1-2-3 spreadsheet. The interface provided is a set of 1-2-3 menus that integrate directly into 1-2-3. Users may use this to read from Oracle databases into spreadsheets, write from spreadsheets into Oracle databases, and embed SQL commands into 1-2-3 spreadsheet cells. A similar product for WordPerfect is already available on the VAX and will probably be ported to the PC world this year. Many more product linkages can be expected in the coming years.

SQL — The Language

In taking a quick tour of the SQL language, we will use a sales order form as our point of reference with the application world (see Figure 1). This form will be implemented as four tables: *customers*, *items*, *orders*, and *orderlines*.

SQL as a language is conveniently divided into three components: DDL, DCL, and DML. The DDL (Data Definition Language) provides the means of creating and altering the structure of the database. DCL (Data Control Language) is concerned with controlling access to the data in a multiuser environment. And DML (Data Manipulation Language) is the facility for querying, adding, and changing data. This last is the component invariably used by the press when giving examples of how simple SQL is to use.

DDL: Data Definition Language

DDL consists of a set of statements for creating and altering the structure of data tables. Once a table is created and populated, columns may be added or modified, but not deleted. The reasons for this should start to become apparent after our discussion. Let's begin our

9-Track Tape Subsystem

*The Data Interchange
Solution You've
Been Waiting For!*

Qualstar's 1/2" 9-track Ministreamers™ bring full ANSI data interchange capability to your PC or Macintosh™ computer system.



With 9-track tape, you are free to exchange data files with any mainframe or minicomputer in the world.

Our affordable Ministreamers come in both 7" and 10½" versions and use less desk space than an ordinary sheet of paper. They can provide 1600 thru 6250 BPI capability and may be used for disk backup as well as data interchange. Complete subsystem prices, including software, start as low as \$2,495 for 7" units and at \$3,670 for 10½" units.

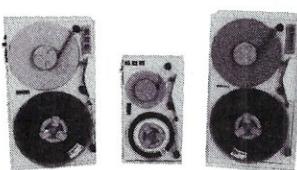
Qualstar has become the market leader in desk-top 9-track systems for a good reason; our tape drives have established an outstanding record for reliability and low cost of ownership.

Discover the many advantages 9-track tape has over other Micro/Mainframe links.

Call us today!

QUALSTAR®

9621 Irondale Avenue,
Chatsworth, CA 91311
Telephone: (818) 882-5822



Macintosh is a trademark of Apple Computer, Inc.

CIRCLE 67 ON READER SERVICE CARD

dp_MAX

dBase III Tools in Turbo Pascal 4.0

Complete Support for dBase III files.
DBF, NDX, DBT file & record Access.
Fully Compatible dBase III B+Tree ISAM.
Library of 100+ functions in TP4 Unit.
Allows 250+ files & indexes open at once.
LRU file caching, round robin file manager.

$$dp_MAX = \int_{-\infty}^{+\infty} dBASE\ III +$$

Turbo Pascal 4.0

Max Software Consultants, Inc.
4101 Greenmount Avenue
Baltimore, MD 21218
(301)-323-5996

\$149

CIRCLE 97 ON READER SERVICE CARD.

tour of DDL by looking at the types of data elements we have to work with.

- **Data Elements.** What are the basic data types supported by SQL? While different implementations may provide variations, the basic data types are *CHAR*, *VARCHAR*, *LONG*, *NUMBER*, and *DATE*.

CHAR and *VARCHAR* are used to represent character strings, with *VARCHAR* representing variable length strings. In some implementations, *CHAR* and *VARCHAR* will be treated identically. The maximum length of the string must be declared, and most implementations will have a maximum limit of 240 to 255. A 20-character field, such as a last name, would be declared as either *CHAR(20)* or *VARCHAR(20)*.

LONG is used to represent variable length blocks of text and has an implementation-specific maximum size in the thousands of characters. Most implementations place extra restrictions on *LONG* columns, so it should not be viewed as equivalent to *VARCHAR*. In particular, *LONG* columns are usually restricted from being part of the search criteria in queries.

Different implementations may sponsor a plethora of numeric types in addition to the generic *NUMBER*. *NUMBER* is adequate by itself in most cases, unless a significant performance benefit is available. Vendor documentation should always be checked carefully to see if this is the case. The keyword *NUMBER* can be used by itself to designate an integer, or it may be used with a length and an optional decimal length. It is similar to declaring a numeric field in dBASE III.

Declaration	Meaning
<i>NUMBER(4)</i>	4-digit integer
<i>NUMBER(7,2)</i>	7-digit number with 2 decimal places (99999.99)
<i>NUMBER</i>	integer with system-defaulted length

Care should be taken in using system-defaulted lengths, as these can be quite large (i.e., 40 digits!). The vagaries of *DATE* tend to be implementation-specific in terms of the default format and the range permitted. *DATE* can be used as a data type, since this then allows date arithmetic to be used conveniently in queries.

- **Relations.** Relations or tables are the basic building block of SQL databases. They contain the actual data that users want to store and manipulate. The basic command for creating a new table is, simply enough, *CREATE TABLE*. The data type for each column is specified as part of the table definition.

Let's start by defining the table *customers* for our sales order form. To make things a little easier to sort out, we'll keep the SQL keywords in upper case.

```
CREATE TABLE customers (
  custid  CHAR (5)      NOT NULL,
  name    CHAR (25)      NOT NULL,
  addr    CHAR (25),
  city    CHAR (15),
  state   CHAR (2),
  zipcode CHAR (5));
```

In this example, the keywords *NOT NULL* mean that the *custid* and *name* fields may not be left empty. (Empty is a possible value for all fields, even numeric.) For numeric fields, empty or null is different from having a value of zero. You can thus always tell whether a field has an intentional value of zero as opposed to having never been filled in.

We are going to want to access the *customers* table by the *custid* column frequently. Knowing this, we will set up a permanent index on that column. This is done by the statement:

```
CREATE UNIQUE INDEX custindex ON customers
  (custid ASC);
```

The above *CREATE INDEX* does a lot. The keyword *UNIQUE* indicates that duplicate values are not permitted in the *custid* column. The keyword *ASC* next to *custid* indicates that the column is to be kept sorted in ascending order. *DESC* is also available to indicate descending order.

We will also want to access this table by name (nobody will have their customer ID handy when they call).

```
CREATE INDEX custname ON customers (name ASC)
  COMPRESS;
```

This time, we do not insist on uniqueness since the same customer may have multiple accounts. We also introduce the keyword *COMPRESS*. *COMPRESS* indicates that the value of *name* is not to be copied into the index, pointers will be maintained back to the values of *name* in the *customers* table. This last option gives up a little speed but saves a lot on space.

Our accounting department informs us on seeing the above definitions that *custid* is not long enough for their customer identification scheme and that they want to keep track of a credit limit and total unpaid purchases in the table as well. We humor them and make the changes by:

```
ALTER TABLE customers MODIFY (custid CHAR (8));
ALTER TABLE customers ADD (credit NUMBER (8,2)
  NOT NULL, purchases NUMBER (8,2));
```

Note that adding and modifying are accomplished by two different commands. Once again there is no provision for deleting columns.

The *orders* table will contain one row or tuple for each sales order. We will not keep any information in this table from the *customers* table except for *custid*. The *custid* column will thus relate corresponding rows of the two tables. One of the principles of designing relational databases is to avoid keeping duplicate copies of data whenever possible. Since it is possible to infer *name*, *addr*, *city*, *state*, and *zipcode* from *custid*, there is no reason to duplicate the values of the other fields in the *customers* table.

One of the artificialities of relational databases is also based on the same principle. We would like to be able to

place any number of different inventory items on the same sales order. To accomplish this, we split the items being ordered into a separate table: *orderlines*. This is necessary because SQL has no provision for repeating groups of columns within a table; very few database systems can handle that type of construct with finesse.

```
CREATE TABLE orders (
  ordernumber  NUMBER (5)      NOT NULL,
  custid      CHAR (8)        NOT NULL,
  initials    CHAR (3)        NOT NULL,
  orderdate   DATE);
```

```
CREATE TABLE orderlines (
  ordernumber  NUMBER (5)      NOT NULL,
  stocknum    CHAR (7)        NOT NULL,
  quantity    NUMBER (3)      NOT NULL);
```

Note in these commands that the column *ordernumber* appears in both tables and is used to relate them. *custid* also appears in the *orders* table and is used to associate the individual sales order with the customer name, address, and credit information from the *customers* file. We are still missing several items from our original sales order form shown in Figure 1: the item description and price columns, for instance. Since we hope to sell each item many times, there is no need to constantly repeat this information, and we place it in a separate table containing only information about the items being sold.

```
CREATE TABLE items (
  stocknumber  CHAR (7)      NOT NULL,
  description  CHAR (35)     NOT NULL,
  qoh         NUMBER (4)      NOT NULL,
  price        NUMBER (7,2)    NOT NULL);
```

There is one last thing we need to consider before our table definitions are complete. Since we are using one of the better implementations, we have *CLUSTERS* available for our use. In DDL, a *CLUSTER* is a means of informing the database manager that two or more tables are so closely related that their physical records should be combined into one physical file. For instance, every time we access *orderlines*, we will also want to access *orders* in order to get the appropriate header data. It would save time during both queries and data updates if the corresponding rows of the *orders* and *orderlines* tables were collocated on the disk drive (avoiding disk movements can speed an operation up by six to 10 times on the average). This is exactly what a *CLUSTER* command does. The tables will be sorted on one or more cluster columns,

Figure 1. Sales order form for sample database

Order #:	99999	Shipping Address		
Customer ID:	XXXXXXX	Name:	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	
Taken By:	XXX	Addr:	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	
Order Date:	99/99/99	City:	XXXXXXXXXXXXXX	XX XXXX
Qty	Stock #	Description	Price	
999	XXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	99999.99	
999	XXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	99999.99	
:	:	:	:	:
999	XXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	99999.99	
			Subtotal	999999.99
			Sales Tax 6%	99999.99
			Total	999999.99

SOFTWARE ENGINEERS Join Our Engineering Team

Nanometrics is a leader in computer-based measurements and inspection systems for the semiconductor industry. We are currently looking for a Software Engineer to join our team of engineers working on highly automated light optical inspection stations and scanning electron microscopes.

The ideal candidate will have the ability to specify, develop, test and document software under UNIX* for PC-based target systems. A BS or MS in Computer Science or equivalent and at least 5+ years' experience as a Software Engineer, with 1+ years in a project management capacity are required. You must have 3+ years of experience programming in "C" and at least 2 years working in an MS-DOS environment. You should be familiar with machine control software, graphics-oriented user interfaces, mathematical algorithms, and the UNIX development environment. Strong written and verbal communication skills are a must.

Nanometrics offers a challenging work environment and a competitive benefits package. If you qualify, please send your resume to: Personnel Administrator, Nanometrics Inc., 690 East Arques Avenue, Sunnyvale, CA 94086. EOE

*UNIX is a trademark of AT&T Bell Labs.

NANOMETRICS

CALL ADVERTISER DIRECTLY

Celebrating
our 10th
Anniversary!

AMX 68000

KADAK's
engineers bring years
of practical real-time experience
to over 600 installations world-wide.

**This real time
MULTITASKING KERNEL
simplifies real life
product development**

- No royalties
- Full source code included
- * C interface
- Preemptive scheduler
- Intertask messages
- Terminal Handler
- * Dynamic memory allocation
- * Event Manager
- * Semaphore Manager
- * Resource Manager
- * Buffer Manager

AMX 68000 operates on any 68000/10/20 system

Manual only \$75 US
AMX 68000 \$1600 US
* Options — ask for price list.
(Shipping/handling extra)

Also available for 8080, Z80,
8086/186/286.

KADAK Products Ltd.
206 - 1847 W. Broadway
Vancouver, B.C. Canada V6J 1Y5
Telex: 04-55670
Fax: (604) 734-8114
Telephone: (604) 734-2796

CIRCLE 104 ON READER SERVICE CARD.

and rows with the same cluster column key will be physically stored together.

```
CREATE CLUSTER ordercluster (ordernumber);
```

This creates a cluster, but does not assign tables to it. That is done with the *CREATE TABLE* command. We would change the commands originally used to create *orders* and *orderlines* as follows:

```
CREATE TABLE orders (
    ordernumber      NUMBER (5)      NOT NULL,
    custid          CHAR (8)        NOT NULL,
    initials         CHAR (3)        NOT NULL,
    orderdate        DATE) CLUSTER ordercluster
    (ordernumber);
```

```
CREATE TABLE orderlines (
    ordernumber      NUMBER (5)      NOT NULL,
    stocknum         CHAR (7)        NOT NULL,
    quantity         NUMBER (3)      NOT NULL)
    CLUSTER          ordercluster (ordernumber);
```

Since clusters have to be created before tables, it becomes necessary to delete tables in order to cluster them. Of course, there will be other operational necessities for deleting tables, but this is as good an excuse as any to introduce the *DROP TABLE* command.

```
DROP TABLE orderlines;
```

When a table is dropped, all the data it contains is deleted, the structure is deleted, and any associated indices are automatically deleted. Corresponding commands are available to delete a cluster or to delete an index without touching the associated table.

DML: Data Manipulation Language

We now set aside DDL and begin to look at what makes SQL famous — its data manipulation facilities. Please note that I do not mean data entry facilities, which for the most part are available only through the tools we'll examine later. For convenience, we will subdivide DML into three areas: queries, *ad hoc* report generation, and creating and updating data.

• Queries. Once you have become accustomed to SQL query facilities, you will be limited only by your ability to formulate queries. You will see what I mean in a moment.

To generate a simple listing that contains all the data in a table, one simply enters:

```
SELECT      *
FROM       items;
```

This will produce a listing of all the items in our inventory. The * in the *SELECT* clause says to use all columns in the table, and the *FROM* clause identifies which table to use. This simple query thus says list all columns from table *items*.

To become slightly more sophisticated and get a listing only of the stock number, description, and price, we would enter:

```
SELECT      stocknumber, description, price
FROM       items;
```

The listing would now omit the *qoh* column ("quantity on hand"), and we would have a simple price list. Of course, the sales force would like to have a list of just the expen-

sive items (they are on commission after all and don't try to move the cheap stuff). We oblige and create a list of items that cost more than \$1,000.

```
SELECT stocknumber, description, price
FROM items
WHERE price > 1000;
```

We have now introduced the *WHERE* clause, which allows us to place conditions on which rows are selected by the query. The possible variations of *WHERE* clauses can form an article all by themselves. In addition to the usual comparison operators (*>*, *=*, *<*) and boolean operators (AND, OR, NOT), various set operators are available, and *WHERE* clauses can also be nested. Let's look at some additional examples. The code

```
SELECT custid, name
FROM customers
WHERE purchases > credit AND state = 'NJ';
```

selects all customers from New Jersey who are over their credit limit. And

```
SELECT quantity, stocknumber, description,
       quantity * price
  FROM orderlines, items
 WHERE orderlines.stocknum = items.stocknumber;
```

creates the table for our original sales order form and also illustrates a simple join. Here we are drawing information from two different tables, and using the *stocknum* column from the *orderlines* table and the *stocknumber* column from the *items* table to decide which rows from the two tables should be combined. Also notice that the last item in our *SELECT* clause is a calculation.

Suppose we want to calculate the net sales for September. This is accomplished by:

```
SELECT SUM (quantity * price)
  FROM orderlines, items, orders
 WHERE orderlines.stocknum = items.stocknumber
   AND orderlines.ordernumber =
        orders.ordernumber
   AND (orders.date BETWEEN '09/01/88' AND
        '09/30/88');
```

Here we have combined information from three tables using the first two parts of the *WHERE* clause. The set operator *SUM* is used in the *SELECT* clause to add up the values of all the rows selected, and the operator *BETWEEN* is introduced as the last part of the *WHERE* clause. All beginning programmers want to use *BETWEEN* operators in the *IF* statements of the first procedural language they learn; finally they have a language that supports it. The output from this query will be a single number which is the gross sales for September 1988!

Let's look at just a couple more examples, starting with the remaining parts of our sales order form. To calculate the subtotal we use:

```
SELECT SUM (quantity * price)
  INTO memvar
  FROM orderlines, items, orders
 WHERE orderlines.stocknum = items.stocknumber
   AND orderlines.ordernumber =
        orders.ordernumber;
```

The *INTO* clause is used to place the result into a memory variable. This is an implementation-dependent feature,

Locate C Bugs BEFORE they Bite with PC-lint

PC-lint will analyze your C programs (one or many modules) and uncover glitches, bugs, quirks, and inconsistencies. It will catch subtle errors before they catch you. By examining multiple modules, PC-lint enjoys a perspective your compiler does not have.

PC-lint is full K&R plus ANSI extensions. It will find argument/parameter mismatches, inconsistent declarations, uninitialized variables, unaccessed variables, suspicious macros, unreachable code, indentation irregularities, function inconsistencies, unusual expressions, printf-scanf format irregularities, and much, much more. All error messages can be turned off locally and globally.

"... a remarkable well thought-out product which will check for just about every conceivable coding error ... Its value increases with frequent use ... we confidently recommend PC-lint."

Andrew Binstock, The C Gazette

"PC-lint has everything going for it: flexibility, speed, good documentation, and a reasonable price. I exercised the product daily on a large, working, project to see if I could include it in my development tools. The answer is a definite YES."

*Stephen D. Cooper, Blue Notes
San Francisco PC Users Group*

"... friendliness is a prime consideration in PC-lint. Gimpel has provided ways to make Lint shut up about all those errors you either know or don't care about. If Unix-Lint was implemented as well, I would use it more."

Don Malpass, IEEE Software

NOW AVAILABLE -- Generic Lint in shrouded source form for use on VAX/VMS, Unix, Xenix, Microport, Versados, IBM VM/CMS - MVS, OS-9, etc. Requires only K&R C to compile. Prices start at \$798. Call for details.

Gimpel Software

3207 Hogarth Lane
Collegeville PA 19426
(215)584-4261

PRICE: \$139.00 first copy, \$100 each additional
VISA, MC, COD -- PA residents add 6% sales tax
30 Day Money-back Guarantee -- Specify MS-DOS or OS/2
Works with any MS-DOS C compiler - direct support for 12
C compilers including Microsoft, Turbo, Lattice, Desmet
PC-lint and Generic Lint are trademarks of Gimpel Software.

CALL ADVERTISER DIRECTLY

especially regarding the manner in which queries that return multiple values are handled. If the implementation you are using supports this, then sales tax and grand total are a snap based on the value of *memvar*. If not, you can always use:

```
SELECT .06 * SUM (quantity * price)
FROM orderlines, items, orders
WHERE orderlines.stocknum = items.stocknumber
AND orderlines.ordernumber =
orders.ordernumber;
```

for the sales tax and

```
SELECT 1.06 * SUM (quantity * price)
FROM orderlines, items, orders
WHERE orderlines.stocknum = items.stocknumber
AND orderlines.ordernumber = orders.ordernumber;
```

for the grand total.

...the buyer must beware since not all implementors have adopted the standard, especially among the existing microcomputer databases that are migrating toward SQL and where partial implementations abound.

Suppose, in order to help you decide where you might put some warehouses, you want a count of how many orders were shipped to different states. This can be accomplished with:

```
SELECT state, COUNT (*)
FROM customers, orders
WHERE customers.custid = orders.custid
GROUP BY state;
```

Here we introduce two more elements of DML, the *COUNT* set operator and the *GROUP BY* clause. *COUNT*, as you may have guessed, simply counts the number of rows that have been selected. *GROUP BY* provides a means of telling the database manager to aggregate the data on a particular field.

We can't leave this topic without showing you a nested query:

```
SELECT custid, name, addr, city, state, zipcode
FROM customers
WHERE custid IN (SELECT custid
FROM orders
WHERE date < '06/01/88')
ORDER BY zipcode;
```

This query will create a mailing list of all customers who placed one or more orders prior to June 1, 1988. Each customer will appear only once regardless of how many

orders were placed, and the list will be displayed in zip code order.

This discussion has given only a brief overview of the query capabilities of SQL. There is not enough room here to cover the many other query functions contained in the DML.

- **Reports.** In many cases, all you need to create a complete columnar report, in addition to the query, is some way to organize a page heading and column titles. Interactive implementations of SQL usually supply some means of doing this. Oracle, for instance, provides a *COLUMN* command to generate column titles, *TTITLE* and *BTITLE* to supply page headings and footings, and a *BREAK* command to specify page breaks on a change in column value, and to blank out column values that are identical after the first one on a page.

Together with the query facility, these commands permit the user to create fairly sophisticated columnar reports without the aid of a procedural programming language. This is very valuable in the typical organization, where application backlogs are such that programming resources are not available to help construct *ad hoc* reports.

- **Creating and Updating Data.** Before breezing through the data with all these neat query and reporting features, it really is necessary to add some data. The command for adding a new row to a table is *INSERT*.

```
INSERT INTO items
VALUES ('ABC-123', 'Widget, goldplated', 100,
50.72);
```

As you can see, the basic form of the command is to name the table being inserted into and supply a list of values for the table's columns. The columns may optionally be named specifically. But, you ask, why make people list the columns for the table? As usual, there is more than one form and use for the command. If we wanted to make a table containing only our high credit limit customers for the vice-president of marketing to use to hound them, we could do so by using:

```
INSERT INTO salesleads (custnum, company,
creditlimit)
SELECT custid, name, credit
FROM customers
WHERE credit > 5000;
```

The assumption here is that another table, called *salesleads*, has been previously supplied by a *CREATE TABLE* command and that it has columns similar to those of the *customers* table, but with different names.

In the same way, once our overly enthusiastic sales force has made the sales and the leads turn out to be deadbeats, we can update their credit status appropriately, either individually

```
UPDATE customers
SET credit = 300
WHERE name = 'Honest Abe Used Cars';
```

or as a group

```
UPDATE customers
SET credit = 300
WHERE purchases > credit;
```

The *SET* clause can accept expressions as well as constant values:

386 HummingBoard

TWO PROCESSORS ARE BETTER THAN ONE



Prices Start at just \$1,595

When it comes to 386 add-in boards, the HummingBoard is in a class by itself. It is a true coprocessor that makes productive use of the base system's original 8086, 286 or 386, running it in *parallel* with the HummingBoard's 386.

You can adopt a whole new PC strategy.

Put your demanding applications that need more speed or memory, like 123, dBase, AutoCad, compilers, etc., on the HummingBoard while your network, Desqview, disk cache, TSR's and other applications run concurrently in the *other* 640k on the "base" system.

Our new Second-Shift™ software permits running a time consuming compile, report or other program

totally in the background on the Hummingboard while you continue to use your base system for other tasks.

Turn 640k into 900k

Another HummingBoard benefit is more memory for DOS programs, *without changing your software*. On the HummingBoard, many DOS applications can address from 700k up to 900k directly, without EMS. EMS 4.0 is supported as well.

Blazing performance

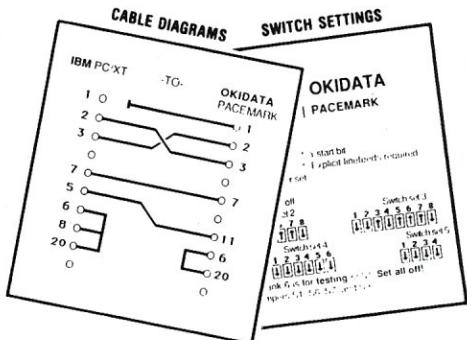
With 16, 20 and soon 25Mhz versions, plus 32-bit cached memory for no-wait-state performance, the HummingBoard is at the top of the class for fast 386 systems.



A.I. Architects

One Kendall Square, Cambridge, MA 02139
(617) 577-8052 FAX (617) 577-9774

INTERFACING MICROS



- RS-232-C
- CENTRONICS
- IEEE

Distributors Welcomed!

Made Easy with MICRO-MATCH

A unique product and service for interconnecting

- Micros to Printers
- Micros to Modems
- Micros to Plotters
- Micros to Terminals ...

To hook-up any two of these devices, *MICRO-MATCH* provides:

- Pin-to-Pin Cable Designs
- Jumper Configurations
- Switch Settings
- Interface Instructions.

Prices begin at \$59 per volume. Send your business card or call for ...

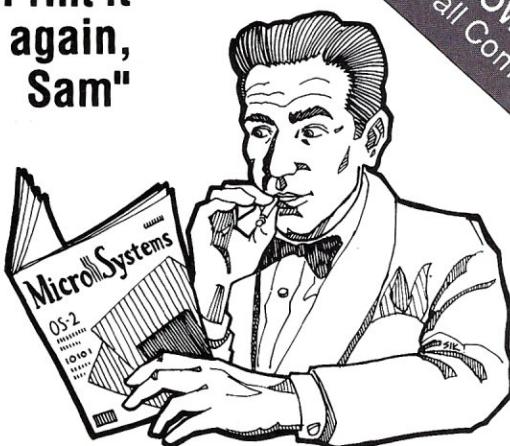
FREE SAMPLES!

COMMAND
COMPUTER CORPORATION

345 Boulevard
Hasbrouck Heights, N.J. 07604
201-288-7000

CIRCLE 96 ON READER SERVICE CARD.

"Print it again, Sam"



ORDER NOW!
for Fall Comdex

Spread the good news about your product with a Micro/Systems Reprint

You can have your article or review professionally reprinted on 60 lb. paper stock, in 4-color, 2-color or black & white. Call today for more information.

Shaun Hooper
(415)366-3600

* Minimum quantity 500 reprints

CIRCLE 114 ON READER SERVICE CARD.

```
UPDATE customers
SET credit = .5 * credit
WHERE state = 'NJ';
```

Updates may also be performed using the results of a query:

```
UPDATE items i
SET (qoh) = (SELECT qoh - SUM (quantity)
  FROM items, orderlines
 WHERE items.stocknumber =
 orderlines.stocknum
   AND items.stocknumber = i.stocknumber
 GROUP BY stocknum);
```

This last one is a bit complicated but illustrates the update power of DML. The object is to decrement the inventory quantity by the quantities sold. The *SELECT* command groups together the *orderlines* rows based on *stocknum* and sums the quantity field for each group. This amount is then used to decrement the *qoh* column from the *items* table.

You can see from this that interactive data entry is not exactly the strong point of SQL. It is more awkward than a form-based system for updating individual records, and perhaps provides a little too much power for updating groups of records. For this reason, a variety of add-on products is available for creating interactive applications based on SQL database engines. These products generally offer an interface that lacks query power but offers a considerable improvement in the data entry area.

• **Advanced DDL: Views.** Having developed the preceding database structure, the accounting department now informs us that they do not want the sales clerks to have access to the credit information contained in the *customers* table. The DDL portion of SQL provides us with a means of doing this, without reorganizing the data, via *views*. A *view* is, in simple terms, a means of creating virtual tables. For instance, we can accommodate the request made by accounting through the command:

```
CREATE VIEW clerkcustomers AS
SELECT custid, name, addr, city,
state, zipcode
FROM customers;
```

There will now appear to be a table called *clerkcustomers* in the system that is identical to *customers* except for the two missing rows. Whenever *customers* is updated, *clerkcustomers* will appear to be updated automatically. We can use a view to combine information from two or more tables:

```
CREATE VIEW shipto AS
SELECT ordernumber, name, addr, city,
state, zipcode
FROM orders, customers
WHERE orders.custid =
customers.custid;
```

Notice the *WHERE* clause is used to connect the two tables. In essence, the full functionality of the *SELECT* statement can be used to create views of the database. These views can encompass multiple tables. Further, once created, a view can be treated just as a normal table in terms of the *UPDATE* command, and can be used as the object of an *INSERT* command as long as all the *NOT NULL* columns of the underlying tables are part of the view.

When an *UPDATE* command is applied to a multitable view, the appropriate columns in the underlying tables

are updated. Views can be a powerful tool for creating simplified environments for less well-trained employees, as well as for simplifying the application of data security commands, and as a macro facility for capturing queries that are used often.

DCL: Data Control Language

SQL was a multiuser environment from its inception. It thus contains commands for controlling access to data, as well as the normal assortment of locking commands. In general, locking is handled automatically in the interactive environment and is not a concern to the user.

The basic security commands provided for a database administrator (DBA) are *GRANT* and *REVOKE*. As you can undoubtedly guess, *GRANT* gives access and *REVOKE* takes it away. Continuing with our sales order system, if user Cromby is in charge of the inventory department, the DBA might give him full authority over the *items* table, including the ability to *GRANT* or *REVOKE* access to other users by:

```
GRANT ALL ON items TO cromby WITH GRANT OPTION;
```

Cromby may decide that anyone should be able to read the table and issue the command:

```
GRANT SELECT ON items TO PUBLIC;
```

In addition to granting *ALL* privileges, or just the *SELECT* privilege which allows reading but not changing a table, the DBA or his or her designees may grant *ALTER*, *DELETE*, *INDEX*, *INSERT*, and *UPDATE* in any combination. Additionally, the *UPDATE* option can have a list of columns:

```
GRANT UPDATE (addr, city, state, zipcode) ON
customers TO jones;
```

This command does not give Jones the ability to *UPDATE* the *custid*, *name*, *credit*, or *purchases* columns. Effectively, it allows Jones only to perform address corrections.

In addition to operating on tables, *GRANT* can be used to bestow *DELETE*, *INSERT*, *SELECT*, and *UPDATE* privileges on database views. The combination of *GRANT* with the view facility effectively gives the DBA the ability to control data access at the level of individual columns and rows.

At a higher level, *CONNECT*, *RESOURCE*, and *DBA* authority are available. When granting these, the user password is also required. *CONNECT* provides the basic ability to log on to the database, *RESOURCE* permits the creation of database objects, and *DBA* gives access to full DBA authority.

```
GRANT CONNECT, RESOURCE TO user IDENTIFIED BY
password;
```

This is the basic command required to add a new user to the database from the SQL side. Individual implementations may impose additional non-SQL procedures on the process.

The *REVOKE* command operates analogously and never requires a password to be specified.

```
REVOKE CONNECT FROM user;
REVOKE UPDATE ON items FROM user;
```

This concludes our whirlwind tour of the SQL language. Many commands, clauses, options, and functions haven't

TRAQWER™ ... Keeps Track...

- Of C Functions
- Of C Structure and Union Definitions
- Of C Complex Data Objects

And Lets You Find What You Need at the Touch of a Key

SYSTEM REQUIREMENTS: MS-DOS or PC-DOS, 2.0+; at least 256K; CRT with ANSI.SYS or monitor with ROM-BIOS support.

SUPPORTS: MDA, CGA, EGA, VGA, MCGA, HGA
HARDWARE INDEPENDENT

MS-DOS is a trademark of MICROSOFT CORP., PC-DOS is a trademark of IBM CORP.

\$150⁰⁰

N.J. residents
add 6% sales tax.

tropic software 

168 WOODBRIDGE AVE., #2, HIGHLAND PARK, N.J. 08904 • (201) 572-0515

CIRCLE 115 ON READER SERVICE CARD.

even been mentioned. It is a very rich and powerful language for the manipulation of databases.

SQL Environments

Just as important as the functionality of the SQL language are the facilities for using it in a practical environment. What are the typical support components found in an SQL package?

Interactive SQL

The entire discussion thus far has centered on an interactive SQL interpreter. The user sits at a command mode prompt and interactively enters commands and has results displayed, just as when he or she is at the DOS prompt.

The typical architecture for an SQL system is to have a database management kernel to which all the other components interface. Interactive SQL is thus a user-system interface for direct use of the kernel by an end user. Command scripts that are typically supported are similar in structure and function to .BAT files in MS-DOS.

SQL will shield the user from locking tables. In general, the interactive environment is friendly and productive for a user who has taken the time to become comfortable with SQL. Since the chief advantage of the interactive environment is the facilities for *ad hoc* queries, *ad hoc* reports, and global table updates, this is not the best of all possible worlds for the data entry staff.

Report Generation

SQL report generators come in two flavors: the screen painter variety and the report language variety. Both varieties allow record selection criteria based on embedded SQL queries. The screen-painter approach is more

common among the PC packages that are migrating to SQL; the report language is more common among the SQL packages migrating from mainframe or minicomputer environments, although some of these have already announced plans to release screen painter-based products.

The report language approach can be very rich in terms of the structure of the reports generated. This approach has the additional advantage of having a source file the user can look at that shows all the operations being performed in generating the report. It has the disadvantage that these report programs can become quite long (3000 statements is not unheard of) and hopelessly complicated to maintain.

The screen painter approach benefits from the what-you-see-is-what-you-get approach, but suffers the same problem as some application generators in terms of how to document what is present in the report definition thus created.

4GL Facilities

SQL is a classic example of a nonprocedural language. There are no constructs available for control flow. What facilities are provided that enable developers to create procedurally oriented applications? What fourth-generation language (4GL) approaches outside of SQL are merged with it?

Ingres is an example of a system that has implemented a procedural language to complement SQL. When creating an application using Ingres, the developer embeds SQL statements within the procedural language programs. This approach can be appreciated by programmers who are more accustomed to dealing with procedural languages. It also avoids some of the self-documentation pitfalls of the application generators, but it suffers from the productivity bottlenecks associated with procedural language development. Ingres is a fairly good example of the approach, providing both a form definition facility and provision for a query-by-example type of interface.

Menu and Application Generators

SQL vendors are among the players in the field of programmerless programming. How successful are they in this regard? What approaches are being taken? How much customization outside of the application generator is realistically required?

Oracle took an approach different from Ingres, initially going with a pure SQL approach. The company has since backed off from this and is including a procedural language in its latest DEC product. There has been no announcement as to when, if ever, this will appear at the PC level.

Oracle's approach to an SQL-based application generator is a product called SQL*FORMS. This product provides the developer with a screen painter for forms—dataentry and query-by-example style reporting use the same form—and a mechanism called a trigger. A trigger is a point in the processing of the form where the developer can attach SQL statements for execution. These points include the depressing of a function key, before or after a field is entered, after a field is changed, before a form is entered, and at several dozen other processing points. SQL statements can be inserted at any trigger point. Additionally, a simple macro language is provided to allow forms to call other forms, manipulate memory variables, and reprogram function keys.

The approach is novel and seems to lend itself to rapid development, but it is potentially a maintenance nightmare. The package does not come with a good facility for documenting the applications created.

A Reliable PC/XT Compatible for the Cornerstone of Your Products

SLICER Announces The SLY40-XT.

The SLY40-XT is a small (4-1/4" by 9-1/4") four layer card featuring all of a PC/XT mother board functions. This board is composed of just 17 low power CMOS ICs, and it simply plugs into a passive back plane or SLICER's TEN SLOT BUS BOARD.

- NEC's 8 MHZ V40
- One Megabyte of Zero Wait State RAM
- Ideal For Tough Industrial, OEM and Portable Applications
- American Made and Fully Supported by Slicer

PC and XT Are Trademarks of International Business Machines

MasterCard
Visa
Check
Money Order
C.O.D.



Slicer Computers Inc.
3450 Snelling Ave. So.
Minneapolis, MN 55406
612/724-2710
Telex 501357
SLICER UD

CIRCLE 112 ON READER SERVICE CARD.

Interfacing to 3GLs

All of the above facilities may seem fine to you, but can you use SQL through calls from your favorite programming language? The answer for most implementations is yes. As usual, though, there are several issues to consider.

It is possible to write portable programs that contain direct calls to SQL engines. In attempting to do so, however, you should restrict yourself to using the facilities that are part of the ANSI SQL standard and/or IBM's System Application Architecture (SAA) Common Programming Interface (CPI) for SQL. There are now a number of IBM publications for developers that specify parts of the SAA interfaces. The relevant one for SQL is IBM document number SC26-4348-0. The decision to comply with SAA will restrict you to using one of the "official" SAA languages, which currently are C, COBOL, and FORTRAN.

If SAA doesn't grab your fancy, a call interface is generally also available. These are typically much lower level than the SAA interface and are not standardized. Since the SAA interface is typically a preprocessor that translates down to this level anyway, the developer is much better off dealing at the SAA level. In fact, you should probably not consider using products that do not provide an SAA interface.

Regardless of the type of interface you choose, you should be very reluctant to go to the third-generation (3GL) level. Unless you are doing substantial number crunching or specialized string processing—which are questionable applications for SQL databases anyway—all the facilities you need should be present at higher levels. The one valid consideration that may drive you to this level is performance. Seriously question if the additional performance is worth the additional development and maintenance costs associated with what is essentially multiple-language programming.

CASE Support

The SQL implementors coming from the minicomputer world are already embroiled in a competition over who will provide the most CASE (Computer Assisted Software Engineering) support most quickly. At this stage, these packages are some variation on the theme of either data dictionary or data flow diagram support. Several vendors have tools that will generate completed applications from CASE-level interfaces at various stages of completion. Whatever else you can say about these tools, you can't call them cheap; typical starting prices are in the \$2,000 range (and remember that does not include the DBMS).

This is still a turbulent area and it is unclear how useful the present tools are, or how soon full-scale tools will be available. Because of the potential productivity issues involved, this is definitely an area to investigate when planning a purchase and one to monitor after a purchase is made.

Application Development Considerations

Now it's time to get down to brass tacks—actually putting a system together. While comprehensiveness would require a multivolume book, we can address some of the major "gotchas."

Efficiency

Are there data structures that should be avoided? What types of organization will yield better performance in a practical application?

Relational databases, in general, have a reputation for being slower than their hierarchical and network cousins. This stems from the fact that links between tables are implicit in relational databases and based on physical

pointers in the other two. This, of course, is also the source of much of the flexibility inherent in relational databases.

Most of the SQL implementations include a high-tech piece of software called a query optimizer. Such software usually consumes large amounts of memory and is based on artificial intelligence technology. The bottom line value of a query optimizer to you as a developer is that the order of clauses in the commands will have no effect on performance, and indices will be automatically used if available and applicable. While you are relieved from worrying about whether there is a best way to order the clauses, you are not relieved from worrying about the best statement form to use. In the more complex queries, there will be several different ways to state the same query. Some optimizers will not automatically realize that the set operation you have requested could be done more efficiently as a nested query.

The general principle to keep in mind when designing a database structure is that the movement of the read/write head on a disk will cause significant delays. Any strategy that minimizes head movement will maximize performance. Thus, while a purist's approach to organizing a relational database will tend to result in a large number of tables, each of which contains a small number of columns, this will result in poor performance. The purist's approach will result in users issuing a large number of joins, either explicitly or via views, which in turn will cause the disk head to bounce back and forth between the areas where the tables involved in the joins are located.

One strategy for minimizing such head movement for tables that will be involved together in frequent joins is to use the clustering feature, if it is available. This can improve join performance by a factor of 6 to 10 times;

Attention Graphics Programmers and System Developers

The TI 34010 is the best news to come along in the graphics processor arena in a number of years. PC Tech, Inc. has developed two TMS 34010 based PC compatible display adapters that allow the full power of the TMS 34010 to be used on PC/XT/AT compatible machines. **DGIS** is available for either board.

MONOCHROME

- 256k display memory
- 256k program memory
- Up to 48 MHz 6 MIPS operation
- Both ECL and TTL monitor drivers
- Multiple programmable dot clock frequencies
- Typical resolution 736 x 1024
- Capable of drawing 125 million pixels per second
- CGA / MDA / Hercules Emulation
- Suggested list price \$595 (\$1295 with full page black on white monitor.)

COLOR

- Up to 1 Mbyte display memory (8 bit / pixel)
- 1 - 4 Mbyte program memory
- 256 simultaneous colors from palette of 256,000
- Up to 48 MHz 6 MIPS operation
- Dual (piggyback) single slot board
- Multiple programmable dot clock frequencies
- Typical resolution 1024 x 768
- Custom dot clocks available
- Suggested list price \$1295 (512K video memory, 1 Mbyte program memory)

TI 34010 Assembler, Debugger and "C" Compiler available!

Both boards support **DGIS** from Graphic Software Systems, Inc. Most major text and graphics applications are supported on both boards via **DGIS** and low level graphics interfaces. Custom implementations of either design and private labeling are also available.

Contact:

PC Tech, Inc.
904 N. 6th Street
Lake City, MN 55041
612-345-4555
FAX: 612-345-5514

PC, XT and AT are trademarks of International Business Machines Corp.
DGIS is a trademark of Graphic Software Systems, Inc.
Prices subject to change.

CIRCLE 46 ON READER SERVICE CARD

however, there is usually a trade-off involved in using this facility. To simplify the implementation of clusters, implementors tend to trade space for time and complexity by requiring that each cluster record be an even fraction of the physical block size. Thus, while clusters are faster, they take more room and waste space unless your tables accidentally work out to be just the right length.

Since clusters can have a certain amount of dead space in each physical record, you may feel there is no space penalty—up to a point—for storing redundant data in clustered tables. You may thus want to consider whether using redundant columns in cluster tables will help to avoid joins with tables that are not part of the cluster.

Another strategy to consider involves the availability of multiple physical disk drives. Performance in a join can be improved if the tables involved in the join are not resident on the same disk drive. This is because multiple disk drives mean multiple read/write heads, which of course assumes only one user is contending for the drives—and is an invalid strategy for network servers. If you are thinking about purchasing large capacity drives

In the new database servers, the query is processed in the server and only the results are sent back to the network station. This is a major difference from past systems, where the intelligence was in the network station.

for individual workstations, however, you may want to consider whether performance might not be better served by multiple smaller drives of the same speed.

The final consideration in minimizing disk head movement is to provide either a large amount of RAM for application level buffering, or to provide a large amount of RAM right on the disk controller board for use in disk caching. This, of course, leads us right into the next topic.

Hardware Considerations

How much disk space do you need? Is disk caching necessary? How much RAM? What kind of processor? Will a coprocessor help? What type of backup system? How often?

The answers to these questions will depend to some extent on the context in which the system is used (workstation, stand-alone, or database server). In general, you can count on needing an 80286 processor in an AT configuration with 640K of RAM. Coprocessors are not usually utilized by these implementations, as number crunching is not their forte.

For stand-alone systems, find out how the database managers you are considering manage their disk space. This may seem like an absurd issue to investigate, but Professional Oracle requires that disk space be preallocated to it, and the space becomes unavailable to other applications. If you allocate 20 MB to Oracle, it may

initially use only 5 MB, but all 20 MB are lost to other uses. To recover this space once allocated, Oracle must be reinstalled!

Any system being used in either stand-alone or database-server mode should be equipped with a very fast processor, very fast disk drive, and lots of RAM for use in buffering. Check out your implementation to find out how much RAM it can effectively use. This can have a dramatic impact on performance.

Plan on daily backup. The backup system can be a stream tape cartridge, preferably with the same capacity as the disk drive so that only one tape need be loaded to perform the backup. It will probably not be possible to back up or recover individual database components in the form of tables or indices, so a system that permits random access of the backed-up files for recovery is not needed.

Communications

What protocols do these systems support? What types of LANs are supported? Is special hardware required? What type of network loading can be expected?

The database server products for LAN environments that you are hearing so much about in the press, especially in conjunction with OS/2, behave very differently from the LAN versions of products such as dBASE III. In the new database servers, the query is processed in the server and only the results are sent back to the network station. This is a major difference from past systems, where the intelligence was in the network station. The result is that while the communication load on the network decreases, the processing load on the server increases dramatically. Any machine used as an SQL database server, where only results are sent on the network, should have the fastest possible processor in combination with the fastest possible disk. You will definitely want to consider all the memory the database manager can take advantage of for buffering, and also consider getting a large disk cache capability. As your staff becomes familiar with SQL, the machine(s) on which database servers reside will tend to be the system bottlenecks—no ifs, ands, or buts, about it!

The issue of LAN hardware and software support is more complicated. The database managers that are migrating down from mainframe or minicomputer environments tend to come via UNIX implementations, and consequently will initially support TCP/IP instead of Novell's SPT protocol. TCP/IP usually means Ethernet LAN hardware. The situation will be different for the PC packages, such as dBASE IV and Paradox, which add SQL support. You should check into the issue of protocols early in your selection process. Salespeople for the TCP/IP-based vendors tend to avoid the issue until you are committed to their product.

Distributed Databases

How is a distributed database handled? How invisible can it be made to the end-user?

Any horror can be made invisible to an end user if enough effort is expended by the developer. In the case of distributed databases under SQL, it is neither very horrible nor hard to hide. The *CREATE DATABASE LINK* command allows either a developer or an interactive user to log on to a nonlocal database. Once the link is established, a table on the foreign database is accessed by attaching *@linkname* as a suffix to the table name. This, of course, means that you have to know in advance what machine each table is going to reside on. If a table is later moved from one machine to another, updating all the scripts, report definitions, and application generator definitions could get ugly.

In an interactive SQL environment, the end user can be hidden from the link syntax via the mechanisms of views and logon scripts.

More important to the developer are such issues as how deadlocks are detected and resolved, and how integrity is maintained. For instance, what happens when one of the database servers on the network goes down? These issues are very much implementation- and release-specific, and should be researched carefully before undertaking a cross-machine development project.

On the bright side, you may never find another mechanism for creating a distributed application that is as easy to set up and as seamless as the mechanism provided through the SQL environment.

Application Maintenance

Once an application is built, changes will be requested over time. How easy is it to maintain an SQL-based system?

As an application development environment, SQL is not yet very mature. Although some application generators will allow you to construct an application very quickly, as the application grows, the process of maintaining it—or even doing the initial construction—can become quite cumbersome. This is due to the fact that application generators often do not come with adequate support in terms of reports documenting the structure of the application generated, or cross references to help identify all the places where a column is accessed or updated.

From the maintenance point of view, an application generator that lacks such facilities should be used only for prototyping. In essence, that is what such a package is best suited for. In some cases, add-ons are available that provide a data dictionary capability or even a full-blown CASE capability. Such packages tend to be expensive, but will pay for themselves during the first year of application construction and can reduce the later life cycle costs of a system by orders of magnitude.

Loading the Data

If you are involved in cutting over an existing database to a new SQL system, one of the problems involved will be how to salvage all the data you already have. Your major worry here will be whether to keep the old schema or not. In either case, the simplest approach will be to create tables that mirror your present structure, move your current data into these tables, and then use the DML to reorganize the data if desired.

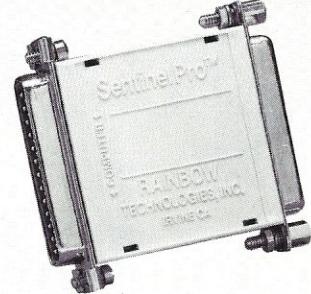
The process of transferring the data will usually involve outputting your present data into a set of flat ASCII files. The ASCII files are run through a utility that converts them to the format used by your new SQL database manager. That means you need at least as much free space left on your system, after you install SQL, as is taken up by your present data files (assuming that you back up and delete the old files after you create the flat ASCII file version of them).

You may be spared the intermediate step if you are converting from dBASE III, since many of the relational databases provide direct support for reading dBASE III format files. Don't count on that being the case, however; they do not all support dBASE III. The availability of direct conversion support from your present database is one of the things you should check out during the database selection part of the process.

In any case, count on the conversion taking more time and more disk space than you originally planned on. □

Did you find this article particularly useful?
Circle number 3 on the reader service card.

Hard Locks for Soft Parts.



At Rainbow Technologies, we think protecting software developers' investments is very serious business. That's why we designed the first fully effective security solution for software running on PCs and other computers.

Our family of virtually impenetrable Software Sentinel hardware keys provides the highest level of software protection the developer can get. While remaining invisible to the end user.

Take a look.

Key Sentinel Family Features.

Prohibits unauthorized use of software □ No need for copy protection □ Unlimited backup copies □ Virtually unbreakable □ Pocketsize key □ Transparent operation □ Transportable

Software

Sentinel.

- Runs under DOS and Xenix, on IBM PC/XT/AT and compatibles
- Algorithm technique (Never a fixed response)
- Serial or parallel port version
- Minimal implementation effort
- Higher level language interfaces included
- 100 times faster than fixed-response devices (1ms)

Software

Sentinel-W.

- Higher level language interfaces included
- Runs under DOS on PC/XT/AT and compatibles
- Parallel port version only

Software

Sentinel-W.

- Designed for workstations, supermicros and minicomputers
- Serial port only (modem-type)
- Algorithm technique
- We provide detailed interface specifications: Developer creates a port driver
- Interface requirements: 25 pin DB25P or DB25S; RS232/RS422/RS423
- Only signals used: DTR & RTS from computer; signal ground; DSR or optional DCD from Software Sentinel-W or external device. TXD, RXD, CTS, RI passed through.

Call For Software Sentinel Evaluation Kit Pricing.



RAINBOW TECHNOLOGIES

18011-A MITCHELL SOUTH
(714) 261-0228 TELEX: 386078
IRVINE, CA 92714 USA
FAX: (714) 261-0260

CIRCLE 95 ON READER SERVICE CARD

Structured Query Language and Its LAN Alternatives

by William Wong

*A comparative look at
SQL and some
competing database
products available now
for local area networks.*

The Structured Query Language (SQL) and SQL LAN servers are hot topics this year. SQL is the standard program level interface to a relational database. But what is SQL? How does it fit in with relational database products? Is SQL really a standard? How does SQL fit into networks? And what are the alternatives?

Database architectures did not start out with relational database technology. Flat files came first and are still with us. Flat files were followed by indexed files using hashing- or Btree-access methods. Most flat-file systems now support one or more indexed methods to speed retrieval. Hierarchical and network database architectures are more sophisticated than flat file systems and have emerged as the workhorse for large database application. In fact, most mainframe database applications are built on IBM's IMS (Information Management System). Accessing data from these database systems normally requires an application program to navigate through the database. The relationships between different items in the database, like all the students in a class, is specified when a database is built. The layout and manipulation specifications for these database systems tend to be built into application programs. *Ad hoc* queries or reports often require the creation of new programs.

Relational database systems tend to be more flexible, both from the user's and programmer's point of view. User-accessible database query programs tend to be a standard part of any relational database product. Query By Example (QBE) is a standard methodology used by many of these products. QBE lets you select the fields you want to look at and specify the criteria to be used for

William Wong is president of Logic Fusion, Inc., a systems software development firm located in Morrisville, Pennsylvania.

selection. Hierarchical and network systems can have QBE front ends, but it is a more difficult task to build the front end because much of the information about how to access the various fields in the database is kept within programs.

The technology for relational database systems was developed in the late sixties. Dr. Edgar F. Codd, a mathematician and former IBM fellow, published a paper entitled "A Relational Model of Data for Large Shared Data Banks" (*Communications of the ACM*, June 1970). The paper is the basis for subsequent work on relational database design. An early relational database language was SEQUEL (the Structured English Query Language), which was developed by IBM's Research Laboratory. SEQUEL/2 changed into SQL. More than 10 years later IBM introduced SQL/DS, which has been transformed into DB2, IBM's mainframe relational database system.

A number of major companies have relational database products ranging from IBM's mainframe-based DB2 down to PC-based products. Some products run on more than one system, such as Oracle, which runs on PCs, minis, mainframes, and LANs. Relational database architectures are now the products of choice for many applications.

PC-based database products are almost as numerous as word-processing products. This article examines some products from the various database categories, including DataFlex from DataAccess Corporation, dBASE from WordTech Systems, and Professional Oracle from Oracle Corporation. These systems all run on single-user, network, and multiuser systems providing an applications developer with a host of options.

DataFlex is a relational database system with sophisticated form and report systems that is not based upon SQL. DataFlex also runs on a number of non-DOS computers. dBASE is a dBASE III super clone that is a relational system and a superset of the popular dBASE product from Ashton-Tate. Professional Oracle is a complete relational database system that uses SQL. Oracle also runs on non-DOS computers, including major mini- and mainframe computers.

Database Architectures

A flat-file database system may sound like a simple product but, properly implemented, it can satisfy a large number of application requirements. A flat-file database usually consists of fixed-size records with fixed-size fields. This can be augmented with variable-length records and fields. Index files or sorting utilities are normally used to speed retrieval. Flat-file products range from menu-driven to sophisticated programmable systems. In fact, one of the most popular PC database products, dBASE, is effectively a flat-file database system with some relational support tacked on to link files together.

A hierarchical database system is similar to the tree-structured directory system found in DOS. Records can contain data fields like a flat-file system and fields that are links to other records or sets of records. The links are defined using a data definition language, like that found in a relational database system, but in a hierarchical database the links are made directly to the related records. Direct linking can speed access to related data but makes it more difficult to update and maintain.

A network database system is similar to a hierarchical database system but a network database system is more flexible. It also requires more sophisticated programming methods, which is why network database systems tend to be embedded within applications. Links within records can be arbitrary in a network database, which makes them very efficient as well as flexible. However, getting to some information within the database may require a complex

traversal of record links.

In its simplest form, a relational database is a dictionary and a collection of tables. Each table contains rows that consist of one or more fields or columns. The form and content of a column is defined within the dictionary. Tables are related to each other by columns that are common to the related tables. Table 1 lists the standard operations using tables.

Relational database systems are not always as simple as the architecture and functions indicate, although relational systems tend to be easier to understand and build applications for than hierarchical and network database architectures.

A user or programmer views a relational database as tables with an arbitrary number of rows. A relational database designer views tables as columns of various types. The designer is concerned with building relationships between tables. Normal forms are designs and specifications that eliminate redundant data and keep the database consistent based upon rules included in the dictionary. Improper design may allow redundant and possibly inconsistent information. For example, two tables could have *ProductID* as a column type. A problem can arise if both tables also have another column with the same type, like *ProductName*. Imagine two rows with the same *ProductID* but different *ProductNames*!

The relationships between tables can be grouped in normalized forms. First normal form allows any group of relationships to exist. It is possible to have redundant and conflicting data with a database system whose relationships are only in first normal form. At the other end of the spectrum is the fourth normal form. A database system that fits into fourth normal form will not have redundant or logically conflicting data.

Conversion between normal forms is a topic for college courses and seminars and is beyond the scope of this article. There are even tools from companies such as Oracle that assist in database design. In general, simple relational databases can be easily examined to determine whether redundant or conflicting data can be entered.

Figuring out whether a database system is really relational can be difficult. Many products that say they are relational database systems have only some aspects of a

Table 1. Relational Database Functions

Selection:	Select a set of rows from a table.
Projection:	Select a set of columns from a table.
Product:	All rows made by combining columns from two tables.
Union:	All rows appearing in either or both tables.
Intersect:	All rows appearing in both tables.
Difference:	All rows which appear in only one of two tables.
Join:	All pairs of rows from two tables which satisfy a condition.
Divide:	Requires a single column table and a double column table. The data type of first column of the second table must match the data type of the only column of the first table. The new table contains one column whose type matched the second column of the second table. The new table contains all rows where the first column entry of the second table can be found in the first table.

relational database. E. F. Codd developed 12 rules for determining if a product is truly a relational database (see the box). The technical report can be obtained from the Relational Institute, 6489 Camden Avenue, #106, San Jose, CA 95120.

Access and manipulation of a relational database is done using a database language. Although the language does not have to be SQL it probably is. dBASE fits some of Codd's rules, but it falls short in critical areas, such as integrity constraints and data independence. dBASE is also a record-at-a-time language for many options and indexes must be maintained by application programs.

What is SQL?

SQL is a language that can be used to access a relational database. However, SQL is a relational database programming language, not a general programming language. For example, SQL does not define program control statements such as FOR, WHILE, or IF-THEN-ELSE. For this reason, SQL is often found embedded in existing language implementations such as COBOL or C. SQL can be embedded in a number of ways from incorporation of SQL statements and syntax into the compiler to a single library function that accepts SQL statements as character string parameters.

SQL is really two separate but related languages. The Data Definition Language (DDL) is used to describe and create a database while the Data Manipulation Language (DML) is used to access a database. Both languages are available for use by application designers and users if SQL is available to them. DDL is also used to create new logical views of the database.

DDL and DML have a number of common portions. For example, the first two SQL statements in the following table definition are DDL statements while the third is a DML statement.

```
CREATE TABLE COMPANY
  ( COMPANYNO      NUMBER NOT NULL
  , COMPANYNAME    CHAR(10)
```

```
    , COMPANYADDRESS  CHAR(40)
  ) ;

CREATE VIEW COMPANYNO10 AS
  SELECT  COMPANYNAME, COMPANYADDRESS
  FROM    COMPANY
  WHERE   COMPANYNO = 10 ;

SELECT COMPANYNAME FROM COMPANY WHERE COMPANYNO = 10 ;
```

The first statement creates a new table with three columns where the first column must contain a value. The second statement creates a new view of the same table that can be manipulated just like a table. The last statement creates a new table also, but one which is not incorporated as part of the database. Each resultant table can be manipulated using DML statements.

SQL defines the syntax and semantics for SQL statements and how they operate on a relational database. How these statements are entered or incorporated into an existing language implementation is not part of SQL. One of SQL's advantages is the limited number of operations needed to do real work.

For example, the SELECT statement is used as a stand-alone statement as well as part of other statements, such as when a VIEW is created. The general format for the SELECT statement is:

```
SELECT      columns
FROM        tables
WHERE       condition
ORDER BY    order
```

The SELECT statement is actually used to implement all the database operations listed previously. INSERT, DELETE, and UPDATE statements have a similar syntax and, like all SQL statements, operate on one or more tables within the database.

All SQL implementations are not equal. There is an ANSI version of SQL, and IBM's DB2 is another *de facto*

Codd's Twelve Rules for a Relational Database

To be a relational database:

1. All data must be kept in tables.
2. All items in the database are accessible by using a combination of tables, primary key values and columns.
3. Null values must be supported.
4. The database description should look like normal data that can be manipulated using all the tools usable on normal data.
5. The database system must support at least one language that can be expressed in character strings and provide comprehensive support for the following items:
 - a. data definition
 - b. view definition
 - c. data manipulation
 - d. integrity constraints
 - e. authorization
 - f. transaction boundaries

6. The DBMS must have a method of determining if data can be added or deleted based upon a view, not just a table.
7. All database manipulations must be logically atomic.
8. Changes to storage representation or access methods will not logically impair application programs.
9. Changes to tables and relationships which preserve information will not logically impair application programs.
10. Integrity constraints must be part of the database.
11. A relational database has redistribution independence.
12. A low-level, record-at-a-time language, if available, should not be able to circumvent integrity constraints.

standard that is more complete. The various implementations tend to differ in minor features, implementations, and, more commonly, enhancements. Porting applications from one SQL-based system to another is similar to moving between C or COBOL compilers; keep within the standard and things tend to port without many problems. Use a particular non-standard feature and porting can cause headaches. The problem is compounded when SQL is embedded with other languages or support products, such as a forms or report system, which are nonstandard between vendors.

Some Database Products

DataFlex and Office Works

DataFlex is a sophisticated relational database system that runs on a wide variety of local area networks and computers, including UNIX systems and DEC VAX computers. The package comes with two binders that include a user's manual, an encyclopedia of commands, and a two-part tutorial that steps through creation of an application.

DataFlex is usable by non-programmers for database creation and use but it is really a programmer's tool. It includes a database creation facility, menu creation facility, a report generator, and a sophisticated programming language that is unique to DataFlex. Being unique is not really a problem, even if you do little programming with DataFlex, because most of the tools are really code generators so a report can be customized by modifying the generated program. Unlike SQL, the DataFlex programming language is a complete programming language suitable for application development with control flow statements, keyboard, and screen handling, and support for conventional files as well as database files.

DataFlex is not a full relational database as defined by

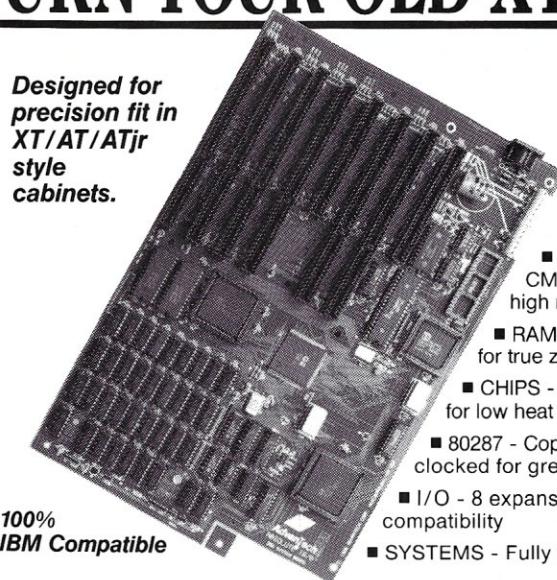
Codd, and it does not use SQL or table-oriented commands like *Project* or *Join*. However, these commands can be implemented using the programming language. Also, the report generation facility handles multiple file relations without the need to write a program. DataFlex does provide validation and verification functions for use with data entry options. Display support includes multiple window support and built-in business graphics support.

The multitasking and network capabilities of DataFlex are limited to standard file locking although DataFlex locks all active files at once. A reread function is used to ensure that the most up-to-date information is available for use. Reread has an implicit lock included with it. The reread function is used because of DataFlex's record-at-a-time nature.

Although the network and multitasking facilities may seem limited compared to other products, DataFlex is quite suitable for developing sophisticated workgroup software. Office Works is another product recently introduced by DataAccess. It is implemented in DataFlex and includes an electronic mail system, an address book, a telephone messaging system, a document control/index system and a sophisticated calendar scheduling system. The electronic mail system includes an editor for creating and viewing electronic mail. The address book is a DataFlex database so you buy DataFlex to augment its use. The telephone messaging system lets someone take phone messages and have the notices sent via electronic mail. The document control system is really a database used to track information that is entered into the database as opposed to a word-processing document control system where the documents can be called up and displayed in a particular format. The scheduling system can be used to coordinate meetings and trips as well as keeping track of critical dates. The sophistication of Office Works shows

TURN YOUR OLD XT_s INTO SPEED DEMONS ABSOLUTELY!!!

Designed for
precision fit in
XT/AT/ATjr
style
cabinets.



100%
IBM Compatible

- CPU - Harris Semiconductor CMOS 80C286-16 for low power, high reliability
- RAM - 1Mb dynamic 60ns memory for true zero wait operation
- CHIPS - Latest THREE (3) AT Chipset for low heat & board dependability
- 80287 - Coprocessor socket independently clocked for greater speed & flexibility
- I/O - 8 expansion slots at 8MHz for true compatibility
- SYSTEMS - Fully tested from \$1595!

**WARRANTY — Full one year on parts and labor,
Nationwide service available.**

Installation Contract Maintenance
Available Through



GE Computer Service

Trademarks:
IBM XT, AT, ATjr - International Business Machines
dBASE - Ashton Tate, Lotus 1-2-3 - Lotus Development
AutoCAD - AutoDesk

AdvanTech Corporation 261 Cedar Hill Street, Marlboro, MA 01752

(508) 481-6009 FAX (508) 481-5215

AdvanTech

that DataFlex is suitable for almost any database application.

dBXL, Quicksilver, and Networker Plus

dBXL is one of a number of dBASE III clone products that are a superset of dBASE III Plus. dBXL is an interactive database product from WordTech that can interpret command files, much like DOS executes batch files, but dBXL is much more sophisticated. Quicksilver is a compiler for dBXL programs that also adds more network support.

dBXL supports a number of relational operations like *Join* and *Set Relation*, but it is not a relational database as defined by Codd and dBXL does not have the data transparency or independence found with SQL. Like dBASE III, dBXL requires indexes and linkages to be expressly stated. Some operations, like *Join* are done on a file (equivalent to a table) but most programs include record-at-a-time operations. As such, dBXL goes past what SQL specifies and provides a complete programming language that is a superset of dBASE. dBXL does have a menu front end for

One of SQL's advantages is the limited number of operations needed to do real work.

creating a database, browsing, and generating simple reports, but the power of dBXL comes from its programming language.

The extensions provided by dBXL make it significantly easier to use than dBASE III, especially when it comes to screen control and local area networks. In addition to windowed text screen control, dBXL includes business graphics support. The files can also be saved for later use by other applications like Ventura Publisher. Network support requires a product called Networker Plus. It allows dBXL to run in multiuser mode using automatic file locking. Networker Plus is included with Quicksilver or as a separate product. Networker Plus also offers a number of other features in addition to providing network support for dBXL.

Networker Plus is a Terminate-and-Stay-Resident program that provides the file and record locking for dBXL and Quicksilver-compiled programs. It must be loaded before these programs are run. Networker Plus is also a pop-up TSR that can send and receive messages from other Networker Plus users on the same network. One additional feature that is unique to database products is the ability to schedule operations to be performed on different computers at a scheduled time for true, although limited, distributed processing. One possible scenario is to have a high-performance PC on the network and all large jobs sent to it, including printing of large reports, sorting large database files, or other batch operations that do not require a great deal of user interaction.

The Quicksilver compiler product is necessary to access the more advanced networking features, but there are additional benefits too, including access to functions written in other languages. Quicksilver is a true compiler that generates object files that can be linked into an executable (.EXE) file. Quicksilver also supports overlays. However, some of the interactive features, like the business graphics support, are lost, but it is still possible to draw graphs that have already been laid out.

Quicksilver network support includes file and record locking on an automatic and programmed level for explicit control. This type of support is found on most database products, but Quicksilver also provides more sophisticated record handling options as well as communication and task options. A SENSERANGE option is provided to determine if another user is working on the same records you are. The communication options include sending and receiving messages from other network users, including the ability to send the contents of the current screen. User identification is also available to Quicksilver programs. The task option is the basis for the distributed processing feature found in Networker Plus. This allows an application to perform its task, then check the task queue for the workstation on which the program is running and execute another program when the application is idle. The application gains control after the other task finishes execution.

Although dBXL and Quicksilver do not run on UNIX or other non-DOS computers, the number of DOS-based computers and DOS-based networks is sufficiently large to make these products appropriate for a large number of users. Networker Plus is a useful tool by itself, but it can be coupled with an application to provide a very powerful workgroup environment. The network and database support provided by WordTech's products make it a very good tool for application development. The ability to link functions written in other languages like assembler or C allow Quicksilver applications to make use of just about any PC hardware available.

Professional Oracle

Professional Oracle is a development product that includes SQL within its database language. It is only one product within Oracle's product line. Professional Oracle comes in three boxes, which include two dozen volumes plus a smattering of handy note cards and booklets. Most of the product names start with SQL such as SQL*Calc and SQL*Forms. Professional Oracle is designed to run as a stand-alone, single-user system on an 80286 or 80386 computer with at least 1.5 megabytes of RAM and a hard disk. It can also work on a local area network which has an Oracle SQL server (more on this later). A companion product, called SQL*Net, is available that runs on 8088-and 8086-based PCs. SQL*Net provides the same operational environment as Professional Oracle but omits the development environment. SQL*Net also requires a connection to an SQL server using either a local area network or some other connection such as a serial link and a modem.

The major pieces that make up Professional Oracle include SQL*Plus, SQL*Calc, SQL*Forms, SQL*Report, and Pro*C. SQL*Plus is an interactive SQL interpreter that executes a language which is a superset of SQL. However, the language is not a complete development language with support for windows and other useful items, such as program control statements. The interface is sufficient for *ad hoc* queries along with database management operations, like setting up a database and managing security. The manual describing SQL*Plus is laid out as a tutorial like most of the Professional Oracle manuals. It effectively teaches SQL by example.

It's Our Objective To Keep You Oriented

*"The only publication in
the world devoted
exclusively to object-
oriented programming."*

Innovative and applicable—tracks and evaluates the latest advances in this methodology.

Presented by a world renown Editorial Board including Adele Goldberg, Brad Cox, Bjarne Stroustrup, Kurt Schmucker and Bertrand Meyer.

Featuring:

Incisive peer-reviewed articles on:

- problem solving techniques
- reusable OOP components
- latest language
- applications in artificial intelligence
- software maintenance
- software management
- program testing

Plus—authoritative columns on Smalltalk, Objective C and C++ languages.

Get unbiased product reviews and up-to-the-minute product news. And personal interviews with the world's top programmers.

Subscribe today!
Order toll free (7 days a week)

**Journal of
Object-Oriented
Programming**

1-800-345-8112

or mail in this coupon

JOURNAL OF
OBJECT-ORIENTED
Programming

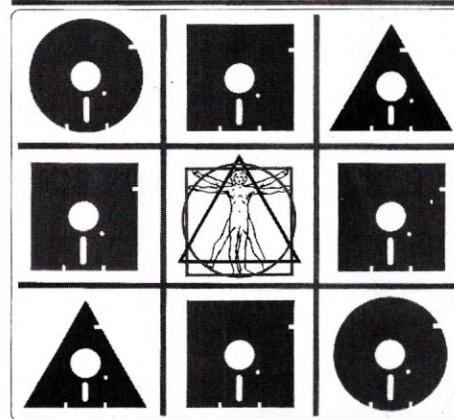
JUNE/JULY 1988

VOL. 1, NO. 2

Coupling an Expert
System Shell with an Object-Oriented
Database System

Designing
Reusable
Classes

Active Objects:
An Access Oriented
Framework for
Object-Oriented
Languages



JOOP

1 year (6 issues) \$49
 2 years (12 issues) \$89
Outside the U.S. add \$24/yr air service.
Check enclosed Bill Me
Charge my: VISA Mastercard

Card # _____ Exp. Date _____

Signature _____

Your Name _____

Company _____

Address _____

City _____

State _____ Zip _____

Mail to: JOOP, 310 Madison Avenue, Suite 503, New York, NY 10017, or
send for a sample copy.

1088

SQL*Calc is a spreadsheet interface to Oracle. It works like Lotus 1-2-3 and many of the commands are identical. SQL*Calc uses the dollar sign, '\$', to prefix SQL commands. Tables can be built within the spreadsheet or kept on disk. For many users, SQL*Calc will be the only thing they need to use. However, there is actually a better alternative if all you need is something like SQL*Calc. Instead of buying the complete Professional Oracle package, you can buy Oracle for 1-2-3. This is a Lotus 1-2-3 V2.01 add-in that lets 1-2-3 manipulate an Oracle database just like SQL*Calc. However, Oracle for 1-2-3 has the same requirements as Professional Oracle, including an 80286 or 80386, 1.5 MB of RAM and 5 MB of hard disk space, which should not be too hard to come by for the power user who needs the capabilities of a high-end relational database. Oracle for 1-2-3 adds a number of built-in functions along with new menu selections. It requires extended memory and 1-2-3 can also use the larger memory for very large applications.

The SQL*Forms and SQL*Report systems are used to create forms and reports, respectively. They both consist of a design tool where the layout is drawn on the screen along with the tool to process the design and present the data on screen or printout. Professional Oracle could be used with just SQL*Calc, SQL*Forms, and SQL*Report, but the program interface is where most applications will come from.

Pro*C is the PC-DOS-based, C language interface to Oracle. Pro*C includes a precompiler which processes SQL statements that are embedded into a C program. The SQL portion of the statement is the same syntax as that used by the other SQL* products. For example, an SQL SELECT statement may look like:

```
EXEC SQL SELECT EMPNO, NAME, ADDRESS
  FROM      EMPLOYEES
  WHERE     EMPNO = :VAR
```

In this case, .VAR is a local C variable. The EXEC SQL prefix is used by the precompiler to locate SQL statements within the C program. This type of statement is referred to as a static SQL statement. Dynamic SQL statements and C-callable SQL functions are also supported, but the syntax of these statements differs from SQL. For example:

```
EXEC SQL EXECUTE IMMEDIATE :string
  oclose ( cursor )
```

The first statement is a preprocessor SQL statement that executes an SQL statement contained in a C string name *string*. The syntax of the SQL statement is the same as a normal SQL statement and does not require the EXEC SQL prefix needed by the preprocessor. The second statement is an example of the C functions available to a programmer. In fact, the preprocessor converts the SQL requests into the appropriate callable functions, which the C compiler then uses to create an application program. In general, the SQL statements are easier to use.

C is not the only language supported by Oracle, nor is PC-DOS the only environment in which Oracle runs. In fact, Oracle started out on minis and mainframes and has only been moved down to PCs with the advent of 80286s and 80386s. Contact Oracle for details on the systems and language supported. Some of the languages include COBOL and FORTRAN while some of the other systems include the DEC VAX and IBM mainframes.

Professional Oracle is a major piece of software. There is even a complete manual dedicated to error messages and codes. As a stand-alone database product Professional Oracle is suitable as an end-user product and a programmer's tool. However, the real power comes through when a PC running Oracle is tied together with an Oracle server.

Oracle is one of the few full-featured SQL relational database implementations that runs on the full range of computers with links between them. SQL*Net, which also runs on an 8088 or 8086, can be used to access remote Oracle databases. Oracle's current offering includes access to mainframe and minicomputers running Oracle, plus a LAN-based server product called LANserver Oracle, which assumes you happen to have SCO Xenix running on an 80286 or 80386 with an Excelan Ethernet adapter. Excelan's adapter comes with a dual-mode driver that allows you to run a Novell NetWare server on the same cable and access it from the workstation. The second logical channel provided with the adapter software is used by Oracle to make a TCP/IP connection to the Xenix Oracle server. If you need more performance or capacity, use a DEC VAX VMS system running Oracle instead. Other NetWare topologies will be supported via the Micom TCP/IP Gateway for NetWare.

Future products from Oracle should make distributed database systems very interesting. OS/2 versions of both the server and client products have been announced.

How Do These Products Fit Into Networks and Non-DOS Computers?

DataFlex uses a network server as a common file system. Office Works is an example of how a powerful system can be built using a standard file server. However, all database traffic from the file server must go through the LAN. This is the same approach used on the multiuser versions that run on Xenix and DEC VMS, but there is significantly less

Version 2.0

XO-Shell™

Pop-Up Utility for Productive People

No matter whether you write programs or reports, XO-Shell will help you hurdle the barriers to working faster and more efficiently with memory resident features:

- **VIEW ANY FILE** and **TRANSFER ANY SECTION** into your editor or to your printer
- **DO CROSS-REFERENCING** without leaving your editor
- **VIEW, COPY and ERASE** files directly from a **SORTED SCROLLABLE DIRECTORY** Display
- **DO SOURCE-LISTING** while in your application
- **INSERT GRAPHICS CHARACTERS** in your text
- **EXTRACT TEXTS** from data file or wordprocessor generated file
- With a single key stroke **RETRIEVE** previous DOS commands, then **EDIT** and **REEEXECUTE** them

XO-Shell is for PCs, XT's, AT's, PS/2s, compatibles

"XO-Shell has won its way into my programming toolkit, and I recommend it."

Kent Porter, Dr. Dobb's Journal

\$79

plus \$5 shipping & handling

Call today toll-free

(800) 635-5011

WYTE CORPORATION
701 Concord Avenue
Cambridge, MA 02138
In MA: (617) 868-7704
Visa, MasterCard

CIRCLE 59 ON READER SERVICE CARD

overhead in moving information from one task to another.

dBXL also uses a network file server, but it only runs on DOS-based networks. The options provided by Net-worker Plus can be tapped by the user and by some creative applications. One possible option is to build a database server. It would be specific to dBXL and all the tools are provided. Normal database operations using a network file server are the same as for products like DataFlex, but a database server could significantly reduce this overhead. However, unless dBXL is running on a multitasking system, it may be difficult to properly handle multiple requests.

LANserver Oracle and Oracle running on minis and mainframes provide multiuser relational database support. Unlike a file server, Oracle, which is an SQL server, keeps the database on the database server and processes the requests there. In general, this is an advantage because it reduces the amount of traffic between the database and the workstation. For example, a *select* operation that scans a table would cause the entire table to be sent across the network, while an SQL server would do the *select* operation locally and only send information to the requesting workstation when necessary. This reduced overhead becomes more significant as the number of SQL statements and tables involved in a procedure increases.

An SQL server can also provide a more secure database than a file server while maintaining flexibility since no data leaves the SQL server unless the server wants to let it out. Integrity and security controls can be implemented to prevent unauthorized access to information or improper modification of the database. One reason an SQL server is more flexible than a simple file server is that an SQL server can give controlled access to certain columns within a table while restricting access to other columns within the same table. Likewise, columns in a table can be used for certain operations but may not be directly accessible. Effectively, a file server only provides security controls at the file level while an SQL server provides column level security plus integrity controls.

The goal regarding SQL servers being stressed these days is that a single vendor's SQL server can be used as a back-end service for many different front-end SQL requesters. The transactions between the front-end and back-end network nodes would utilize some common protocol, such as SQL statements, sent as character strings. At this point, SQL servers have been implemented, but the front-end systems tend to be implemented by the same vendor such as Oracle.

SQL servers do not have to be linked to a workstation via a LAN. An alternative approach is to link the workstation directly to a mainframe or mini using a terminal connection. Oracle can use 3270 emulation products to directly link to a variety of minis and mainframes.

SQL servers are important because they provide an efficient way to distribute database processing and storage resources as well as provide access to data shared by many users. SQL servers will also provide access to data on machines of dissimilar architectures in a transparent fashion.

Alternatives to SQL database servers include products that run on various LAN servers. Novell supports Btrieve as a NetWare server-based process. Btrieve is a low-level, high-performance, database programming interface, not a combination development/user-level package like the products presented here. OS/2 products running with LAN Manager could perform the same function as Btrieve.

Summary

SQL is a language to define, create, and manipulate a relational database. SQL can be used in a stand-alone

fashion but tends to be incorporated into existing languages, like C or COBOL, or hidden by report and query applications. As such, SQL is not really a language but a common sub-language. This is also why you will not find SQL compilers but rather SQL precompilers.

SQL is important because it provides a common link between relational database products. Learn SQL and you can utilize most SQL products. SQL also provides a link between dissimilar language products accessing a common relational database since SQL is used as the access method.

SQL servers and links to the servers are just starting to become common. Oracle and other SQL vendors have a proven track record and have real products available now with more to come. It is only a matter of time before SQL servers on LANs and mainframes are as common as shared hard disks.

Is SQL worth using? It depends upon the application. Btree's, hierarchical and network database products still have valid uses and are more efficient than relational database systems for many applications, especially if the database is embedded within an application. DataFlex and dBXL are two examples of alternatives to SQL. These products will not be displaced by Oracle in the near future but they will have to contend with it and other products like it when it comes to links to minis and mainframes, as well as LAN database servers. ■

Did you find this article particularly useful?
Circle number 4 on the reader service card.

Product Information

DataFlex Revision 2.3 \$1,250
Office Works \$1,395

DataAccess Corporation
14000 S.W. 119th Avenue
Miami, Florida 33186
(305) 238-0012
Circle reader service #258

dBXL \$599
Quicksilver \$599
Networker Plus \$259
(when purchased with *Quicksilver*) \$99

WordTech Systems
P.O. Box 1747
Orinda, CA 94563-9947
(415) 254-0900
Circle reader service #259

Professional Oracle \$1,295
SQL*Net \$395
Oracle for 1-2-3 \$199
Oracle for SCO Xenix
(single user) \$2,000
(eight users) \$6,000

Oracle Corporation
20 Davis Drive
Belmont, CA 94002
(800) 345-3267
Circle reader service #260

A.I. Architects' OS/386

This 386 operating system provides a platform for 32-bit, protected-mode DOS programming.

by Howard Vigorita

AI. Architects is best known for its 386 HummingBoard coprocessor, which is shipped by Gold Hills Computers with Lisp-based artificial intelligence products. At the heart of these products lies the OS/386, protected-mode, DOS emulation environment.

Several related environments also are marketed by A.I. Architects. OS/386-HB is the version for the HummingBoard. OS/386 is the generic version for single-processor 80386 systems. The CP program, included with OS/386, is an unusual combination of a command processor and a debugger. Also available is OS/286, a protected-mode environment for 16-bit IBM PC/AT compatibles.

A.I. Architects is primarily a hardware and environment manufacturer. The company adapts the 32-bit language products of other vendors for use with its products. It also supplies 16-bit libraries and the EXPRESS conversion utility. These allow the output of a variety of more traditional language packages, such as Microsoft C and Lattice C, to be converted to 16-bit protected-mode operation.

MetaWare Professional 386 Pascal and High C 386 are the established, high-level languages for creating 32-bit protected-mode programs targeted to this environment. These languages were used to write OS/386. Lahey, LPI, MicroWay, and SVS also have 32-bit C, Pascal, and FORTRAN prod-

ucts on the way. For low-level work, OS/386 is compatible with the Phar Lap assembler. For the moment, Phar Lap's linker is required equipment for linking 32-bit programs. However, A.I. Architects and Lahey are jointly developing a linker capable of accepting the 32-bit OMF output by Microsoft MASM. My examination here is from an assembly language vantage point using Phar Lap's assembler and linker.

Programming for Compatibility
Before getting into the particulars of this environment, portability should be considered. It's always desirable to program for as broad a base as possible, even when dealing with a protected-mode environment. Not wiring environment-specific selectors into the application program achieves this end to a limited extent. But there is no plain vanilla approach that will accommodate every 80386 environment. It is possible to detect the operating environment and adjust for its specifics when necessary.

There are two basic ways to identify a runtime environment. MetaWare High C 386 uses an INIT module that examines initial selector values to distinguish between Phar Lap and A.I. Architects. IGC's XAM (originally from Softguard) is also identified by its peculiar initialization of the eAX register. The disadvantage of this approach is that it will fail if future environment enhancements lead to alterations.

A better approach is to use a standardized operating system call. A movement is now afoot among manufactur-

ers of 80386 environments to extend DOS function 30 hex (Get DOS Version). In addition to the DOS version numbers returned in the AH and AL registers, return of a 2-byte OEM environment identifier in the high word of an extended register is being considered. RUN386 returns the bytes "DX" in the high word of the eAX register and OS/386 returns "AI" in eBX. Using this information, it is possible to construct programs that adjust to either environment. (See Listing 1 for an example.)

OS/386 Process Partitioning

Although similar in function to the Phar Lap product, OS/386 is structurally different. OS/386 draws its heritage from its HummingBoard coprocessor roots. Although intended to run on a single-processor system, OS/386 incorporates partitioning of real- and protected-mode procedures. This segregation originated on the HummingBoard where real- and protected-mode subroutines execute concurrently on separate processors. On a single-processor OS/386 system, the main implication of this design is that real- and protected-mode procedures do not normally share a buffer. OS/386 provides two ways to deal with this restriction.

First, OS/386 provides the Real Procedure Call (RPC). An RPC initiates a real-mode subroutine from protected mode. Up to 4K of data can be copied to the real procedure's address space by placing it into a transaction buffer. The RPC facility is ideal for allowing the real-mode processor to conduct video display transfers while a Hum-

Howard Vigorita is an attorney on the staff of the United States District Court in New York and serves as an officer of the New York Amateur Computer Club.

mingBoard concurrently does calculations for the next screen update. Time is saved by having the 32-bit processor copy the transaction buffer memory-to-memory with as few as zero wait-states. The burden of doing the memory-to-video transfer is left to the real-mode processor. This often entails up to five wait-states on the real-mode clock, which at 80386 rates could range from 10 to 15 wait-states.

As a second facility, OS/386 provides for the creation of real-mode windows and segments. In creating a real-mode segment, memory allocation occurs in the real-mode address space. The paragraph address is returned, so the real-mode procedure can address the newly allocated memory. Also returned is an LDT selector for protected-mode access. Creation of real-mode windows is similar to segment creation, except that no memory allocation occurs. A real-mode window can serve as another way to access the video display buffer. Direct video update via a real-mode window is better suited to uniprocessor systems where there is nothing gained by the RPC method of doing two transfers for each video update.

An exotic segment mapping facility is also documented, but not yet implemented. Mapping consists of superimposing a real-mode segment onto the address space of an existing protected-mode data segment. This is intended to facilitate relocating memory-mapped I/O devices into a flat address space. It would be ideal for mapping portions of the video buffer into the address spaces of multiple applications. The main reason mapping is not yet available in OS/386 is because 80386 page translation is not implemented. Page translation remains unimplemented in OS/386, pending resolution of conflicts between 80386 memory management hardware and the 80387 math coprocessor as documented by Intel.

OS/386 DOS Emulation

DOS emulation by OS/386 is somewhat broader than that of Phar Lap's RUN386. In particular, OS/386 emulates the DOS File Control Block (FCB) functions where RUN386 does not. FCBs are also maintained in the Program Segment Prefix (PSP) at their usual location of 5C and 6C hex. The only place that FCB support is absent is in the function 4Bh (exec) parameter block where 32-bit offsets encroach on that space. There is a limitation to using FCB functions under OS/386, however. Multisector I/O requests cannot exceed 16K per FCB call.

OS/386 extends INT 21h pointer

parameters to 32-bits in the same manner that RUN386 does. Unlike RUN386, however, OS/386 retains DOS's memory management granularity of 16-byte paragraphs. This does not impair OS/386's capability of allocating memory for a segment as large as the 80386 maximum segment size of 4

gigabytes.

DOS INT 21h functions not supported by OS/386 are functions 31h (terminate and stay resident) and 58h (get/set allocation strategy). These functions are described in the OS/386 documentation as not being meaningful to protected-mode operation.

Table 1. RUN386 and OS/386 Selectors

Description	Type ¹	RUN386	OS/386
Program Code	R,X	0Ch	0Fh
Program DATA	R,W	14h	17h
Program Segment Prefix	R,W	24h	27h
	R,W	04h	N/A
Video Buffer	R,W	1Ch	1Fh
Environment Block	R,W	2Ch	2Fh
Low Megabyte of Memory	R,W	34h	37h
Weitek 1167 Memory	R,W	3Ch	3Fh

Notes: (1) R = read, W = write, X = execute.

Phar Lap RUN386 Update

Since Phar Lap's RUN386 was last examined here (*Micro/Systems Journal*, June 1988), there has been no abatement to the pace of continued updates. As we go to press, Version 1.2b is being shipped.

Phar Lap no longer offers a memory-resident version of RUN386. The 386 DOS-Extender designation now refers to RUN386 itself.

A most welcome addition is a new 215-page loose-leaf manual devoted exclusively to RUN386. This manual now documents all of the selectors previously reported here, as well as a number of other hard-wired selectors, for complete write access to all page, descriptor, and vector tables. Also documented are supported BIOS calls.

A host of new features also have been added. New functions facilitate taking over hardware interrupts, converting protected-mode addresses to MS-DOS addresses with contiguity checks, managing conventional real-mode memory, loading programs for debugging, aliasing segment descriptors, and changing segment-access rights. A new real-mode procedure call supports mixed-mode programs with data passed either on the stack or in an intermode data call buffer residing in conventional memory. EMS hardware can now be manipulated

with INT 67 hex. Also added are the TSR functions. RUN386, together with a protected-mode program, can remain dormant in memory until activated by a software interrupt or hot key.

A number of new switches have been added to the linker. These control placement of the program in memory, allocation of buffers, and the releasing of conventional or extended memory. This control over buffer placement is a major improvement, allowing RUN386 to attain disk I/O speeds that approach the speed of DOS. Most of these switches can be applied either when linking or when loading the environment. If these new switches are applied to the link, an .EXP file will be generated in a new format. Otherwise, the old format is generated. RUN386 handles both formats.

The assembler has also been updated. For some reason, the SEGMENT AT directive in a protected-mode program now triggers a fatal error. The -NODELETE switch will ignore it, but the better tactic is to use structures instead.

These new features improve RUN386's suitability for writing EMS simulators. Quarterdeck's QEMM Version 4.1 is written with it. RUN386 is specially adapted to run under DESQview with QEMM.

Listing 1. An example program that adapts to either 386 environment

This program detects whether it is running under Phar Lap's RUN386 386|DOS-Extender or A.I. Architect's OS/386 and adjusts accordingly. Note that segments are manipulated, not with absolute values, but with values relative to the code segment for greater portability.

```
; exec.asm execute a program from protected mode
; by Howard Vigorita
; For maximum portability, assemble and link as follows:
; 386ASM EXEC
; 386LINK EXEC -EXE EXEC.PLX
; To execute the DIR command:
;    UP EXEC DIR
;    -or-
;    RUN386 -maxreal FFFFh EXEC.PLX DIR
;-----
cr equ 0Dh
lf equ 0Ah

stack_sz equ 8192
dos equ 21h
bpt equ BYTE PTR

pfp STRUCT
    db 80h dup(?)
    cmd_cnt db ?
    cmd_ln db 79h dup(?)
pfp ENDS

DATA SEGMENT PARA PUBLIC USE32 'DATA'
signon db 'Protected Mode Load & Execute',cr,lf,'$'
mem_msg db 'Error reducing memory allocation'
        db cr,lf
exec_msg db 'Execute failed'
newln db cr,lf,'$'
srch_str db 'COMSPEC='
srch_len equ LENGTH srch_str
program db 80 dup(0)
param_block label dword
env_offset dd 0
env_seg dw 0
arg_offset dd offset arg_buf
arg_seg dw ?
arg_buf db length arg_txt
arg_txt db '/C'
        db 80 dup(0)
sp_sav dd ?
env_id dw ? ; DX=0, AI=3
DATA ENDS

CODE SEGMENT PARA PUBLIC USE32 'CODE'
ASSUME ds:DATA, es:nothing, ss:STACK
PUBLIC exec
exec:
    mov AH, 9
    mov eDX, offset signon
    int dos

    ; detect environment
    ; 
    mov AH, 30h ; get DOS version
    int dos
    rol eAX, 16 ; id in high word
    cmp AX, 'DX'
    jne try_ai
    mov AX, CS ; get code selector
    add AX, 18h ; distance to PSP seg
    mov ES, AX
    jmp short cpy_arg_ln

try_ai:
    rol eBX, 16
    cmp BX, 'AI'
    jne exit ; unknown
    mov env_id, 3

    ; copy command line
    ; 
    cpy_arg_ln:
    xor eCX, eCX
    mov CL, es:cmd_cnt ; byte count
    or CL, CL ; any cmd?
    jz exit
    cmp CL, 1 ; maybe a space
    jne arg_ok
    cmp lpt es:cmd_ln, ' '
    je exit
    arg_ok:
    add arg_buf, CL
```

... listing continues

Also not supported is function *26h* (set program segment prefix). A.I. Architects, like Microsoft, suggests using function *4Bh* (exec) instead.

Although not clearly documented, OS/386 augments DOS emulation with support for IBM AT BIOS calls. I verified operation of many of the INT 10h video functions.

The OS/386 Programming Environment

A.I. Architects recommends environment-independent programming, but OS/386 documents the application-oriented Phar Lap supported selectors. These selectors are modified, however, for a lower privilege level. Under OS/386, user programs are only permitted to operate at Level 3, the lowest privilege level. This makes the environment more secure for program development. The low-level selectors described by Phar Lap as *hardwired*, are not supported.

When OS/386 loads a 32-bit program, it initializes the ES (extra segment) register to point to the PSP in the same manner that DOS does. However, the DS (data segment) register is initialized to the program data area. This change is necessary because there is no support for loading segment registers by reference to segment names in 32-bit mode as is usual in 16-bit DOS programs. This simplifies startup for programs that immediately access data segment variables. However, programmers in the habit of initially processing their command lines with string operations will have to swap ES and DS first.

After initializing segment registers and processing the command line, the next thing most programs do is shrink their memory allocated to the minimum. Because of differences in memory management facilities among environments, this is a good place to take advantage of OEM identification.

As mentioned earlier, OS/386 supports memory management calls identical to those of DOS. This eases converting DOS assembly language code to 32-bit OS/386 calls. But segment arithmetic cannot be used to determine segment sizes. Data, code, and stack segment sizes must be determined with equates and combined with the PSP and environment segment sizes. Although extended function *0EDh* provides segment information, including segment size, it cannot be used to determine data and code segment size either. This is because all available extended memory is automatically allocated at load time, regardless of the MAXDATA linker setting. Note also, that the code and data segments are aliases for one another —

Here At Last!

The Authoritative Source of Database Application and Product Information

Inside DBMS Magazine

DBMS Magazine offers in-depth articles examining the capabilities of programmable databases as tools for solving real business problems. Authoritative features show how a specific product is used to solve a specific problem. Sample some of the feature articles to be presented:

- 50 Nasty dBase Bugs, and How to Squash Them
- Has Ashton-Tate Finally Designed a Decent, Multiuser Database?
- Why 386 Databases Can be Either the Fastest, or a Waste of Money
- Is FoxBase+ dFastest Around?
- Why Microsoft and Ashton-Tate teamed up on SQL: Server – What this Ambitious Product Will and Will Not do for You
- R:Base Puts OS/2 to Work

- Understanding Paradox's PAL
- Can Oracle Squeeze Into a PC?

...with regular columns written by experts, providing tips on developing and modifying databases, such as:

- Putting dBase IV's Arrays to Work
- Cross tabulations in R:Base
- Sharing Files Between 1-2-3 and dBase
- Using Clipper's Valid Statement
- Tips on Creating Reports in Paradox

Who Should

Read DBMS Magazine?

Professional developers, MIS/DP departments, consultants, VARs and serious users will find DBMS Magazine to be the source of important information for serious users of microcomputer database software.

Charter Offer and Benefits with No-Risk Guarantee

For a limited time DBMS Magazine will offer Charter Subscriptions with substantial cash savings.

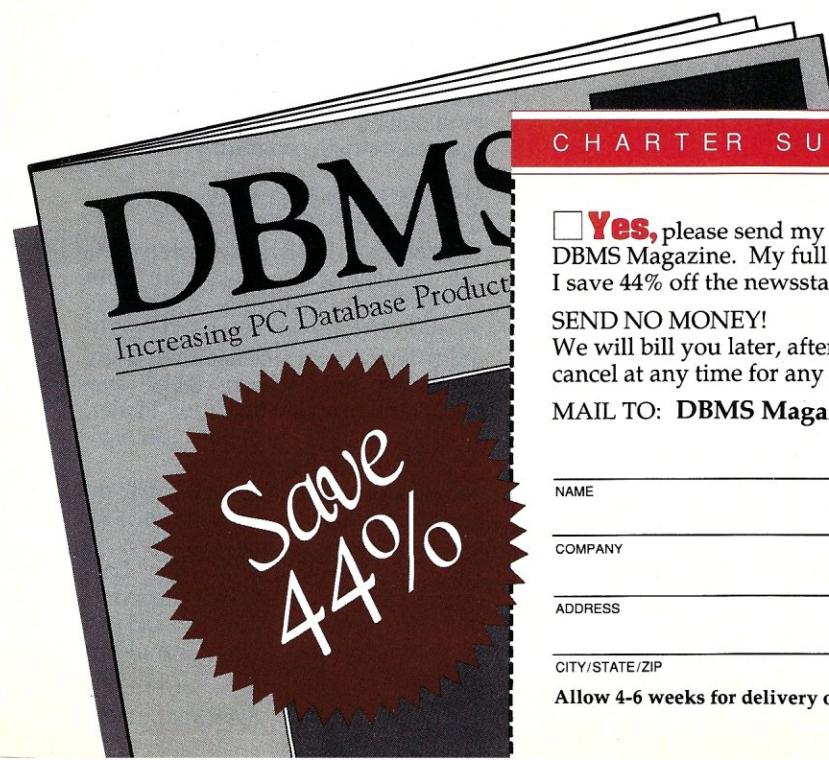
You will also be entitled to continued savings — on all renewals and on any gift subscriptions you may order.

If at any time you decide DBMS is not for you, simply cancel. We will send a full refund for all unmailed issues.

Order Today!

Send your order form to:

DBMS Magazine
P.O. Box 3770
Escondido, CA
92025



CHARTER SUBSCRIPTION ORDER FORM

Yes, please send my first issue and enter my Charter Subscription to DBMS Magazine. My full year's subscription (11 additional issues) is only \$19.97. I save 44% off the newsstand rate.

SEND NO MONEY!

We will bill you later, after you receive your first issue. GUARANTEE! You may cancel at any time for any reason and receive a full refund on all unmailed issues.

MAIL TO: DBMS Magazine, P.O. Box 3770, Escondido, CA, 92025

NAME _____

COMPANY _____

ADDRESS _____

CITY/STATE/ZIP _____

Allow 4-6 weeks for delivery of first issue.

Newstand rate (1 year): \$35

20M7

Listing 1 (continued)

```
; transfer command line to exec cmd buffer
; mov    eSI,cmd_ln
; mov    eDI,offset arg_txt+3
; push   ES          ; swap es & ds
; push   DS
; pop    ES
; pop    DS
; rep    movsb       ; block move
; push   ES
; pop    DS
; ASSUME es:DATA      ; es & ds = data seg

; free up real system memory for child if OS/386
; (Phar Lap uses -MAXREAL switch instead)
free_mem:
    cmp    env_id,3
    jne    srch_env
    mov    eBX,(offset cs:last_byte)+15+stack_sz
    shr    eBX,4
    mov    AH,4Ah
    int    dos
    jnc    srch_env
    jmp    mem_err

; search environment for comspec
; and move it to command line
srch_env:
    mov    AX,CS
    add    AX,20h      ; environment selector
    mov    ES,AX
    xor    eDI,eDI
    mov    eBX,offset srch_str ; eDI = environment base
    mov    eDX,srch_len ; eBX = search string
    mov    eDX,eAX      ; search string length
    xor    eAX,eAX      ; end of string (null)

srch_lp:
    cmp    bpt ES:[eDI],0 ; environment end?
    jz    exit          ; if so, bail out
    mov    eCX,eDX      ; else, eCX = str len
    mov    eSI,eBX      ; eSI = search pointer
    repe   cmpsb        ; block compare
    je    mov_comspec   ; if ZF set, found it
    not   eCX
    repne  scasb        ; else, make eCX big
    jmp    short srch_lp ; & scan for eos
    mov    eBX,eDI      ; save pointer to comspec
    not   eCX
    repne  scasb        ; scan for string marker end
    not   eCX
    mov    eDI,offset program
    mov    eSI,eBX      ; comspec now source
    push   DS
    push   ES          ; swap es & ds
    push   DS
    pop    ES
    pop    DS
    rep    movsb        ; block copy comspec
    pop    DS          ; restore DS

srch_x:
    mov    arg_seg,DS
    mov    arg_offset,offset arg_buf
    mov    eDX,offset program
    mov    eBX,offset param_block
    mov    sp_sav,eSP
    mov    eAX,400h      ; load & execute function
    int    dos
    jc    exec_err

exit:
    mov    AH,4Ch
    int    21h

mem_err:
    mov    AX,CS      ; make sure got data seg
    add    AX,8h
    mov    DS,AX
    mov    AH,9
    mov    eDX,offset mem_msg
    int    dos
    jmp    short exit
    last_byte equ $

CODE ENDS

STACK SEGMENT PARA STACK 'STACK'
db    stack_sz dup(?)
STACK ENDS

END   exec
```

End Listing 1

the same size applies to both.

Ctrl-C interrupt interception is straightforward under OS/386. The standard DOS routine works without modification. The only quirk I noted is that the substituted interrupt handler doesn't gain control as rapidly as the supplied handler. This is evident when Ctrl-C is struck while OS/386 is waiting for a keystroke during function 8h (direct console input). The handler doesn't gain control until detection of the next keystroke (not Ctrl-C). This does not pose a problem under normal circumstances.

OS/386 also provides a pair of extended functions to fine tune interrupt handling. Depending on the type of interrupt, there may be a real-mode handler, a protected-mode handler, or both. Extended function E4h replaces the protected-mode portion

The RPC facility is ideal for allowing the real-mode processor to conduct video display transfers.

and controls calling the real-mode portion before, after, or not at all. Note that this function cannot substitute a real-mode handler. Real-mode interrupt handler replacement under OS/386 can only be done with an RPC, which gains control in real-mode. As an alternative, a TSR can be installed before execution of the protected-mode program.

OS/386's Loader and Command Processor

OS/386 loads itself in a two-step process. Loading a 32-bit program from the DOS command line is done as follows: UP <prog> <args>. The UP utility loads programs and their arguments for uniprocessor 80386 systems. UP contains a bootstrap loader, which in turn loads the OS.386 environment kernel file.

UP can only load two types of 32-bit executable files. A file with an .EXP extension can be loaded if it is in either A.I. Architect- or Phar Lap-defined formats. Note that these formats are different. Although UP can distinguish its own format from Phar

Lap's, symbol files are not loaded correctly unless Phar Lap-linked files are given the .PLX extension.

Also provided with OS/386 is the CP command processor shell. On the surface, CP appears to be a 386 clone of DOS's COMMAND.COM. On initial loading, it executes commands in AUTOEXEC.CP. It supports the full range of DOS built-in commands such as PROMPT, loads 16-bit real-mode programs, and executes DOS batch files. It goes beyond the capabilities of DOS's shell by providing a built-in EMACS-compatible command-line history editor (similar to DOS-Edit). It also allows multiple commands per line and has an ALIAS command that permits redefinition of commands and syntax.

CP's most obvious extension is that it allows running 80386 .EXP and .PLX files from the command line without using UP or reloading the OS/386 kernel. CP keeps the kernel resident, making for brisk protected-mode program loading.

A more innovative aspect of CP is its debugging capability. CP is actually an advanced debugger in the guise of a COMMAND.COM-styled shell. It offers a sharp contrast to the trend toward graphical window-type debuggers typified by Microsoft's CodeView.

One level of debugging extensions which CP provides are additional built-in commands. For instance, LOAD will read a protected-mode executable file and .MAP a symbol file into extended memory. The CODE command does disassembly and the DATA command displays memory contents. With 20 built-in debugging commands, CP has all the capability of Microsoft's Symdeb. In fact, the SYMSYN batch file turns Symdeb commands into synonyms of their CP counterparts.

The potentially most powerful extensions in CP are its active variables and operators. An active variable is a pseudo-variable whose value influences the environment or machine state. The SET command alters an active variable in the same way it operates on environment variables under DOS. The 80386 internal registers are all associated with active variables. The debug registers can be directly manipulated via active variables to give CP full 80386 hardware debugging power.

An extension to the % operator makes CP active variables especially useful. DOS has a special treatment for the % sign in batch files where %1, %2, etc., are substituted for command-line arguments. CP extends % into a general purpose substitution operator. If % is the prefix of a vari-

Power Graphics

Essential Graphics Takes You To New Heights Of Graphic Programming In C. Increases Speed 40%.

When first brainstorming this ad I spent a considerable amount of time trying to determine what graphic image to use as an illustration. The Space Shuttle, Mona Lisa, Robo-Cop - there are so many available.

Then it occurred to me. When you have the fastest, smallest functions, it's really irrelevant to show a complicated graphic image. It would be as if thinking up a sexy graphic were the test of a library.

The Graphics Test

The crucial test of a professional graphics package is: are the functions powerful, reliable, fast, and do they truly eliminate grunt work?

How quickly the functions execute is the criterion most people look for in a graphics library. There is no sense paying for a package that is not up to speed.

Beware Of Speed Traps

We eliminate the bios calls and write directly to the graphics card. As a matter of fact, in a recent benchmark, we were clocked 40% faster than our nearest competition.

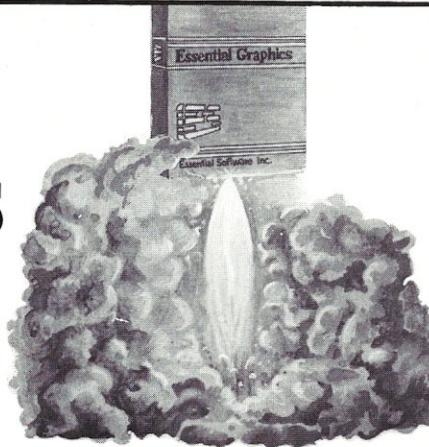
I'd like to repeat that "...clocked 40% faster than our NEAREST competition." Please take a moment to think about the significance of that speed increase in the project you are contemplating or working on now.

Our efficient, granular coding provides you with code sizes up to 75% smaller. Lean, fast and tight - just the way you would have done it yourself.

Power Packed Pixels In Every Package

There has always been a trade-off in this industry between ease of use and power. Our functions do not require a lot of setups, are well-documented, and most of all, thoroughly debugged. Essential Graphics' ease of use stems from our thoughtfulness and not from a lack of power. We explain what we are doing every step of the way. Our support staff consists of the humans who wrote the functions, so we are thoroughly prepared to assist you after the purchase.

Essential Graphics is a trademark of Essential Software



Caveat Emptor

Make no mistake, this is not a package for the "draw a box around the total field" crowd. This library was designed to help the professional C programmer make money and look good.

We've included a complete set of "rubberbanding" functions. One of the most welcome features is the ability to save/restore images in PC Paintbrush format or bit image. World coordinates and view ports aid in programming portability.

We include the ability to manipulate and rotate character fonts and symbols. You can place characters and symbols anywhere on the screen, and use up to eight fonts at one time.

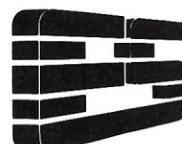
Yours, Mine, Ours

We don't consider ourselves equity partners in your business and therefore we do not charge any royalties or run time fees. We think your efforts belong to you. If for any reason you are unsatisfied with our product you may return it within 30 days for a full refund. Full source is available. Please call today and launch yourself into the world of power graphics.

Price \$299 - Source \$299

Adaptors include - CGA, EGA, VGA, MCGA, ATT, ATT DEB, Hercules, Vega Deluxe, Paradise Autoswitch. **Printer Support** - IBM, Epson, Oki, TI, Alps, Panasonic, and others. **Supports** mice, light pens, plotters, color printers. **Compilers** - Microsoft, Lattice and Turbo-C

Other Essential Products Include: ScreenStar - Essential Communications and Utilities -- /*resident, C*/ - Please call for further information 201-762-6965.



To Order Call:

1-800-451-6174
N.J. 201-762-6965
Fax. 201-762-0118

Essential Software, Inc.

South Orange Plaza
76 S. Orange Ave., Suite 3
South Orange, N.J., 07079

Listing 2. WATCH.CP shell script for OS/386's CP This CP shell script watches for a change in a specified memory word in the DS segment.

```

echo off
IF %EQ %NARGS 2) goto start
echo USAGE: watch [variable]
echo watches for [variable] to change
echo variable may be symbolic
goto done
:start
SET seg=%{DECHEX %DS}
SET value=%{WORD %{seg}:%{${1}}}
echo watching for %{seg}:%{${1}} to change from %value
:loop
step 1 >NUL
if %EQ %{WORD ${DS}:%{${1}} %value} goto loop
STEP 0
:done
:exit

```

• 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 •

Paradox 386
Foxbase+ 386
386-MATLAB./Weitek

... and others ...

These and other protected-mode 32-bit 80386 programs are among the first to take advantage of the full power of the 386. They and practically every **386 protected-mode** MS-DOS program that's shipping were done with MetaWare's compilers.

It's no surprise. The recognized leader, MetaWare introduced the **first C** and **Pascal** compilers that generate protected-mode 386 code for running on any 386 MS-DOS machine (e.g., the Compaq 386 or the IBM PS/2-80) over a year ago. **High C™** and **Professional Pascal™** are well-established and proven.

Smart software developers aren't waiting! Industry leaders such as Borland (ANSA) and Fox use MetaWare's compilers to get dramatic increases in speed and functionality. Don't wait years for Microsoft's 386DOS—your competition will have a big jump on you!

Expand your application to the large 32-bit address space and the full 32-bit registers of the 80386. Go with the long-standing leader. Contact MetaWare for your 80386 software solution today!

(408)429-6382

telex 493-0879

903 Pacific Avenue, Suite 201 • Santa Cruz, CA 95060
The Clear Choice for Large Programming Projects —PC Tech Jnl

© 1987 MetaWare Trademarks Paradox 386 Ansa Foxbase+ 386 Fox Software 386 MATLAB MathWorks High C Professional Pascal MetaWare MetaWare

CIRCLE 56 ON READER SERVICE CARD

able name, its value is substituted. If % is immediately followed by a left brace, as in %{WORD 17:1234}, the expression enclosed by the braces is evaluated and substituted. CP provides a broad range of arithmetic, logical, comparison, and numeric conversion operators for expression formulation. CP expressions use prefix notation similar to that of FORTH and other stack-oriented languages. For example, the expression %{+ 1 2} evaluates to 3.

The lack of a built-in watchpoint facility provided an opportunity to incorporate CP's extensions into a useful batch file. A watchpoint is a type of breakpoint that stops a program when a specified change occurs in a memory location. The simplicity of WATCH.CP (Listing 2) reveals CP to be a powerful, programmable debugger. The faster SETDEBUG.CP batch file, which accesses the debug registers, is supplied on the OS/386 system disk.

Comparative Performance

Comparative Performance
 In order to get some idea of how OS/386 performs relative to Phar Lap's RUN386, I adapted the CRC program for operation under either environment. Table 2 shows the timing results I obtained.

Before placing too much credence on these measurements, however, keep in mind that benchmarking any piece of software can be fraught with problems. Measurements can be very system-dependent. For instance, they will vary depending on the number of DOS buffers and the physical layout of files on the disk. These effects can be minimized for most software by timing operation from a RAM drive. In testing protected-mode environments, timing tests of disk I/O are also dramatically influenced by the location of the buffer. Timings increase substantially when the I/O buffer resides above the one-megabyte boundary. To minimize these effects, I kept the size of the RAM drive at a minimum and placed it in conventional memory.

In the Table 2 timings, OS/386 ran slightly slower than RUN386 on disk I/O. Also, it inexplicably took slightly more time in processing data in memory. This might be due to the fact that RUN386 does memory management on 4K physical boundaries with the help of page translation while OS/386 relies solely on descriptor translation instead. OS/386, however, was able to load almost twice as fast as RUN386, due to its bootstrap design which allows the bulk of itself to be loaded with 32-bit operations instead of the 16-bit COMMAND.COM.

Conclusions

OS/386 seems to be a younger product than its competition from Phar Lap, but it is rapidly maturing. The few bugs I encountered during my evaluation were remedied in a prompt update.

The RPC facilities of OS/386 make it an attractive product for graphics applications when teamed with a HummingBoard coprocessor. Its concurrent processing architecture should enable it to significantly outperform single-processor systems. On my single-processor system, OS/386 performance compares favorably to that of Phar Lap's RUN386.

The debugging capabilities of OS/386's CP are especially noteworthy. When combined with 80386 debugging hardware, CP's command processor extensions result in a hard-to-rival programmable debugger. The power of CP as a debugger is limited only by the programming imagination of the user. My only lament is that CP does not support debugging of real-mode programs.

The 156-page spiral-bound manual of OS/386 is well-annotated with helpful commentary, and documents a number of features currently under development. To supplement a voice-line for technical support, A.I. Architects maintains a BBS stocked with programming examples and updates for download. If OS/386 continues to evolve along the path it has documented, it is sure to become an important trendsetter. □

Did you find this article particularly useful?
Circle number 8 on the reader service card.

Product Information

386 HummingBoard
prices start at \$1,595
OS/386 Development Kit
includes OS/386, LEM
Linker, CP, and EXPRESS \$495
OS/286 Development Kit \$495
OS/286 Evaluation Kit \$50

A.I. Architects, Inc.
One Kendall Square, Building 400
Cambridge, MA 02139
(617) 577-8052

Circle reader service #267

Phar Lap 386 Development Package
includes 386ASM,
386Link, and Run386 \$495

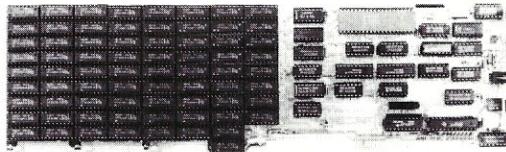
Phar Lap
60 Aberdeen Avenue
Cambridge, MA 02138
(617) 661-1510
Circle reader service #268

Table 2. Real- and Protected-Mode Timing Comparisons

The following are timing comparisons for XMODEM CRC calculations using a table lookup algorithm. Times are listed in seconds using external timer hardware. The CRC is calculated 32 times on a 128K file using a 16-MHz 80386 with all files on a conventional memory VDISK.

	DOS	RUN386	OS/386
Load Overhead	3.3	1.2b	1.914
File Read	.783	1.979	.993
CRC Calculation Loop	3.405	3.722	3.943
	9.339	6.814	7.850
Total Run Time	13.527	12.515	12.786

GET A FLAME



The Blue Flame II is the latest in our line of very high-performance disk emulators for PC's, XT's, AT's, '386's, and all clones. It's extremely fast: 800Kbytes per second transfer rate, ten times faster than hard disks. Even faster than IBM's VDISK program! And big: Up to 8 megabytes per board, 32 megabytes per logical drive. Much bigger than extended or expanded memory. It doesn't waste any of your computer's memory address space for storage. And the Blue Flame II is reliable: With no moving parts, it can be accessed continuously for years with no failures. Don't try this at home with your hard disk!

Not just another RAMdisk, the Blue Flame II has an external AC-powered battery-backup option: Data isn't lost when the computer is turned off. And "Reset" isn't a dirty word anymore. Even during a blackout, the battery maintains data for 10 hours.

The Blue Flame II is available fully-populated, with 8 megabytes, for \$2095. 4 megabytes for \$1195. 2 megabytes for \$795. Battery Backup option costs \$135. Call us for information on our SemiDisk products for S-100, and Epson QX-10/QX-16.

If you want greater software speed, improved data security, increased hardware reliability, get a Flame. If you need the hottest disk performance possible, get a Flame. A Blue Flame II SemiDisk.

SemiDisk Systems, Inc.

P.O. Box 66
Beaverton, OR 97075
(503) 626-3104

CIRCLE 47 ON READER SERVICE CARD.

Using Btrieve with Turbo C and Turbo Pascal

by Tom Hoops and Ed Rought

*Network database
programming
can be tricky,
unless you apply
the right
programming tools
in the right fashion.*

Selecting database management tools for application development can be a formidable task. Since these tools are essential for efficient database manipulation, acquiring the most flexible and expedient tools is important. In this article, we would like to share the experiences that led us to select Btrieve, the database record manager by Novell Inc., as our database development system, and to outline our approaches to using it with Borland's Turbo C and Turbo Pascal.

Many business programs require the use of databases. Developing a database management system from scratch can often more than quadruple the cost, and greatly increase the complexity and unreliability of the program. During the design phases of our first product, a personal scheduling and payroll system, we foresaw this problem and turned to the gurus at Borland International for some affordable alternatives. Enter Borland's Turbo Toolbox, a system of routines written in their popular Turbo Pascal for managing single-user Btree ISAM databases.

Although we underwent some brutal sessions dealing with Toolbox, we eventually adopted it and began integrating it into our first commercial application. To our dismay, as the project neared completion, we realized how application-dependent (less versatile) our routines had become. For example, our routines had to maintain all indexes for the database; this was not a particularly overwhelming chore, but their lack of reusability rendered them worthless for future applications. This problem, along with some other shortcomings, lurked in our nightmares about future projects.

All worries aside, the Toolbox is a wonderful starter package. Providing that you follow the rules and set the page sizes efficiently, the performance of this little package is exemplary. And for a mere pittance. Who can complain? Customers, that's who! "My database got

Tom Hoops and Ed Rought are co-owners of Perpetual Data Systems, a software development and marketing firm in Reno, Nevada. They are also the authors of two books, The Turbo Pascal 4.0 Developer's Library and The C Developer's Library, both published by Howard W. Sams.

squashed—"what do I do now?" is a painful question eventually asked by anyone with a computer. Fortunately, writing a utility that sequentially reads data files and builds new index files to replace the decapitated ones is not a big deal. Nonetheless, when it happened, the task still consumed additional time and money (ours!).

One day, a prospective customer asked, "Does your program run on a network?" Our answer was, "Of course, but only one person can modify the data at a time"—in other words, no. That question didn't seem to be an obstacle until we started asking it ourselves. Additionally, Toolbox has the drawback of allowing only up to 65,536 total records in a database, a limitation we knew we would eventually have to deal with. However, it's hardly fair to condemn Toolbox for these and other limitations considering it is only a light-duty system. Unfortunately, putting Toolbox's disadvantages into perspective didn't make our problems go away. As our interests began to revolve around the amazing world of networks, we knew our infatuation with Toolbox was over.

Writing software for networks is a different animal; there is not much latitude for mistakes in handling data. The companies that own networks expect error-free software, and so do their attorneys. We began playing with the idea of writing our own database management system, but reality clubbed us over the head when our first test run revealed a performance level you would expect of a pocket calculator. Our eyes opened by this revelation, we soon thereafter found our attention caught by an advertisement that said "Be fast, Btrieve" (or something like that). The ad proclaimed that our problems would vanish for only \$595! Before making the investment, we spent a few days conferring with the vendor to be reasonably sure Btrieve was what we wanted.

As we soon discovered, Btrieve has very little in common with our Toolbox. In fact, the only characteristic they share is their conceptual purpose. Therefore, becoming familiar with the new beastly required a bit more effort than we had anticipated. Btrieve is utilized by first calling a program that loads the record manager (database management routines) into memory to remain resident. Then, any application that needs to use Btrieve may issue commands to the record manager via software interrupts. The amount of memory occupied by the record manager is generally minimal; depending on its setup, it can consume as little as 30K.

We have been using Btrieve for more than a year now with Turbo Pascal and, more recently, Turbo C. From our experience, we can say that the DOS Network Version of Btrieve runs like a dream on Novell networks. This version of Btrieve uses the DOS 3.x record-locking scheme rather than Novell's or any other network's mechanisms. As a result, the Network Version of Btrieve not only can be used with Novell's NetWare products, but with any number of DOS 3.x-compatible networks. Btrieve is also available for a single-user environment and is interchangeable with the Network Version, which eliminates the need for changes in the host application's code for this conversion.

Since we have been using Btrieve, we've stumbled across only a couple of drawbacks. The first one deserves a brief history. Btrieve was created and marketed by a company formerly known as SoftCraft Inc. Novell recently purchased the company, presumably because of Btrieve's potential to perform as a background process on a file server and its ability to work well with the company's NetWare product line. Btrieve literature brags extensively about its data integrity through its pre-imaging and transaction controlling. Ironically, Novell's NetWare slaughters this feature with its facility for RAM caching of disk

I/O. If the file server of a Novell network goes down, the file integrity information in the RAM cache disappears, never to be seen again. However, the enormous performance benefits gained by NetWare through its disk caching scheme outweigh the data integrity problem since Btrieve does have facilities for recovering datafiles. Furthermore, Novell recommends the use of power backup systems (UPS) to help prevent such server failures.

The second drawback is more of a courtesy problem between Btrieve application developers. Btrieve's configuration parameters are specified when the record manager is loaded. Therefore, the parameters cannot be changed without first removing the resident record manager (a simple process) and reloading a new one with different parameters. Since only one Btrieve record manager may reside in memory at a time, its host application should remove the record manager from memory when it terminates. If the record manager is allowed to remain, and another application requiring different Btrieve parameters is initiated, the result may be an unpredictable clobbering of the second application—and an extremely unhappy user. Hence, be courteous and write your applications so they remove the record manager when the program is exited.

Listing 1. Global constants, types, and variables

```

const
  MaxFiles    = 20;  {Maximum number of data files to be used}
  MaxKeyLen   = 30;  { Largest key to be used by the system}

  {Btrieve constants}
  Duplicates  = 1;
  Modifiable  = 2;
  Binary      = 4;
  Null        = 8;
  HasSegment  = 16;
  AltColSeq   = 32;
  ModSeg      = 18;  {Modifiable and has segments}
  DupMod      = 3;   {Duplicates, Modifiable}
  DupModSeg  = 19;  {Duplicate, Modifiable and has segments}

type
  FileVar = array[1..128] of char;

  KeySpec = record           {Total size = 16 bytes}
    KeyFos,
    KeyLen,
    KeyFlags : integer;
    NotUsed : array[1..4] of char;
    Reserved : array[1..6] of char;
  end;

var
  FileRecLen : array[1..MaxFiles] of integer; {Record lengths array}
  F          : array[1..MaxFiles] of FileVar; {Main Data file array}
  FN         : integer;                      {Current File Number}
  CurIndKey : array[1..MaxFiles] of Str255; {Last retrieved key}

```

Listing 2. Sample record

```

type
  CustRecType = record
    DateIn   : string[8]; {2} {Actual data offset starting at one}
    ContName : string[15]; {11}
    ContFName: string[15]; {27}
    Position : string[25]; {43}
    CoName   : string[30]; {69}
    Phone1  : string[12]; {100}
    Phone2  : string[12]; {113}
    Address1 : string[30]; {126}
    Address2 : string[30]; {157}
    City     : string[30]; {188}
    State    : string[2]; {219}
    Zip      : string[10]; {222}
    Referred : string[30]; {233}
    NumEmps  : string[5]; {264}
    HrsPerWk : string[7]; {270}
    DLContd : string[278]; {278}
    CustType : Str1;      {287}
  end;

```

Despite these drawbacks, we were pleased with the overall performance of Btrieve. It was only when we first attempted to write a TSR (terminate-and-stay-resident) application that accessed Btrieve datafiles that we became disillusioned with our marriage to Btrieve. (Since Btrieve itself is a TSR, the RAM requirements would have been prohibitive.) Consequently, we began snooping around for a replacement for Btrieve that would work with our C applications and, we hoped, solve our TSR problem. The package we seriously considered was CTree by Faircom. We purchased it, examined it, and promptly returned it (not our usual practice). From our quick analysis of the package, we found no support for automatic index maintenance. Worse yet, the license agreement demanded that we place their copyright notice on our distribution diskettes. The idea of changing our distribution labels while gaining no observable benefits over Btrieve was absurd. Going back to Btrieve, we found a simple alternative to the problem of accessing Btrieve in our TSR application by using a standard record file. Thus, as a result of our brief but educational affair with CTree, we're still happily married to Btrieve.

Now that we've stated our grievances, we would like

Listing 3. Defining a record constant

```

const
  CustFileDef : record
    RecLen,
    PageSize,
    NumIndex : integer;
    NotUsed : array[1..4] of char;
    Variable : integer;
    Reserved : array[1..4] of char;
    KeyBuff : array[1..9] of KeySpec; {Number of keys+segs for each
                                     file goes here}
  end = (RecLen           :698;
         PageSize        :1024;
         NumIndex        :7;
         NotUsed         :(#0,#0,#0,#0);
         Variable        :0;
         Reserved        :(#0,#0,#0,#0);
         KeyBuff[0]      :((KeyPos :2; {DateIn}
                           KeyLen :8;
                           KeyFlags:DupMod;
                           NotUsed :(#0,#0,#0,#0);
                           Reserved:(#0,#0,#0,#0,#0,0#0)),
         KeyBuff[1a]      :{ContLName}
                           KeyLen :15;
                           KeyFlags:DupModSeg;
                           NotUsed :(#0,#0,#0,#0);
                           Reserved:(#0,#0,#0,#0,#0,0#0),
         KeyBuff[1b]      :{ContFName}
                           KeyLen :15;
                           KeyFlags:DupMod;
                           NotUsed :(#0,#0,#0,#0);
                           Reserved:(#0,#0,#0,#0,#0,0#0),
         KeyBuff[2]      :{DLContd}
                           KeyLen :8;
                           KeyFlags:DupMod;
                           NotUsed :(#0,#0,#0,#0);
                           Reserved:(#0,#0,#0,#0,#0,0#0),
         KeyBuff[3]      :{CoName}
                           KeyLen :30;
                           KeyFlags:DupMod;
                           NotUsed :(#0,#0,#0,#0);
                           Reserved:(#0,#0,#0,#0,#0,0#0),
         KeyBuff[4]      :{State}
                           KeyLen :2;
                           KeyFlags:DupMod;
                           NotUsed :(#0,#0,#0,#0);
                           Reserved:(#0,#0,#0,#0,#0,0#0),
         KeyBuff[5a]      :{CustType}
                           KeyLen :1;
                           KeyFlags:DupModSeg;
                           NotUsed :(#0,#0,#0,#0);
                           Reserved:(#0,#0,#0,#0,#0,0#0),
         KeyBuff[5b]      :{State}
                           KeyLen :2;
                           KeyFlags:DupMod;
                           NotUsed :(#0,#0,#0,#0);
                           Reserved:(#0,#0,#0,#0,#0,0#0),
         KeyBuff[6]      :{Zip}
                           KeyLen :10;
                           KeyFlags:DupMod;
                           NotUsed :(#0,#0,#0,#0);
                           Reserved:(#0,#0,#0,#0,0#0)));

```

ADVERTISER INDEX

RSN	Advertiser	Page
89	A.I. Architects, Inc.	31
110	AdvanTech Corporation	41
	Aker Corporation	13
62	Austin Code Works	60
117	Bristol Information Systems, Inc.	15
96	Command Computer Corporation	32
53	Concurrent Controls	67
	DBMS	49
98	Digital Research, Inc.	5
99	Emerald Microware	63
61	Essential Software, Inc.	51
	Gimple	29
105	Hauppauge Computer Works	7
109	Interfirm Graphics Systems	10
	Invisible Software	12
	Journal of Object Oriented Programming	43
104	Kadak Products Ltd.	28
97	Max Software Consultants	26
55	MetaWare Incorporated	58
56	MetaWare Incorporated	52
113	Micronics Computers, Inc.	9
114	Micro/Systems reprints	32
	Nano Metrics	28
	Nanosoft Associates	17
	National Exposition Co., Inc.	71
65	Novell Development Division	2
103	Nu-Mega Technologies	11
46	PC Tech	35
67	Qualstar Corporation	26
86	Quarterdeck Office Systems	C-2
101	Quarterdeck Office Systems	1
95	Rainbow Technologies, Inc.	37
94	Santa Cruz Operation	C-4
47	Semi-Disk Systems	53
112	Slicer Computer	34
81	Stargate Technologies, Inc.	63
51	Sunny Hill Software	72
80	Teletek Enterprises, Inc.	C-3
115	Tropic Software	33
	Wyte	44

ADVERTISING REPRESENTATIVES

<i>National Account Manager</i>	Dwight Schwab (415) 366-3600
<i>National Account Manager</i>	Tami Brenton (415) 866-1957
<i>Advertising Coordinator</i>	Shaun Hooper (415) 366-3600
<i>Advertising Director</i>	Richard Mixer (415) 366-3600

to highlight some of Btrieve's best features:

- Up to 24 keys or segments per data file which are all automatically maintained by the record manager
- Supplemental indexes
- 13 different types of keys (indices)
- Relational access among files
- Support for duplicate, modifiable, segmented, null, and descending ordered keys
- Extended memory
- Partitioned files
- I/O buffer cache
- Integrity controls
- Unrestricted file sizes
- Complete set of utilities for creating and maintaining files

One of Btrieve's better features perhaps is that, unlike Toolbox, it does not include its source code; examining the source code to understand how Btrieve accomplishes its tasks might be a little mind-bending.

The most significant development between Novell's Btrieve and NetWare products is the incorporation of Btrieve into NetWare as a VAP (value-added process).

*Writing software
for networks is a
different animal;
there is not much
latitude for mistakes
in handling data.*

This union can increase the performance of LAN database applications many fold, as well as eliminate the problem stated earlier regarding data integrity, since it is the server—rather than the workstation—that directly processes file requests.

Additionally, with the advent of Xtrieve (Novell's database management system using Btrieve), the end-user can access the datafiles for the purpose of application customization. Also available through Novell is a structured query language system called XQL. This package uses the same memory-resident method of operation as Btrieve, but its extensive memory consumption could prove to be a problem.

We must admit bigamy, though; Btrieve is not our only happy marriage. Borland's Turbo C and Turbo Pascal products have proven to be fruitful partners as well. The programs presented here contain a portion of the source code we use to tie Btrieve in with Turbo C and Turbo Pascal. Not all of the functions available through Btrieve are demonstrated (that would be an extremely lengthy discourse). However, we do try to provide a representative cross section of the essentials. These implementations are particular to our style of usage so, to help you understand them, let us explain our approach.

An interface to the record manager is provided for most popular languages used on the PC today. However, this interface requires a rather long and confusing list of parameters under the Pascal and C methods. To eliminate this and the problem of cryptic Btrieve function specifiers, we use global variables and a variety of functions.

Our typical applications never use more than 20 files, so the global variables that are arrays are only dimensioned to 20 for both languages.

All of the following functions are set up to return the success or error condition reported by Btrieve. FN is primarily used to indicate the current file number to higher-level routines.

Turbo Pascal

In Pascal, we use the global constants, types, and variables shown in Listing 1. An example record would be defined as shown in Listing 2. (Remember, the first byte of a string in Turbo Pascal is its length.) Immediately following the record type definition, we define a record constant, which contains all the information necessary for Btrieve to create the datafile; this is shown in Listing 3. The following is the actual record for moving data back and forth between the application and Btrieve:

```
var  
  CustRec : CustRecType;
```

When we open files, we associate them with an integer. The file number currently in use is stored in the variable FN. The function *BTrv* used in Listing 4 and those following is the Btrieve interface mentioned earlier, which is supplied by Novell. The routine in Listing 4 creates a datafile from the above record constant. Then, the routine shown in Listing 5 opens the file.

To create and/or open the file specified in the *CustFileDef* constant would be done like this (*Stat* is an integer):

```
if (Open_File(F[0], 'Cust.Dat', 0) <> 0) then  
begin  
  Stat := Make_File(F[0], 'Cust.Dat', CustFileDef);  
  Stat := Open_File(F[0], 'Cust.Dat', 0);  
end;
```

Listing 4. Routine to create a datafile

```
function Make_File(var FileB : FileVar;  
                   FileName : Str255;  
                   var FileDef : integer);  
{  
  This function will create file called FileName of FileDef  
  specifications.  
}  
var  
  DumLen : integer;  
  
begin (Make_File)  
  fillchar(FileName[length(FileName)+1], 255-length(FileName), 0);  
  Make_File:=BTrv(14, FileB, FileDef, DumLen, FileName[1], 0);  
end; (Make_File)
```

Listing 5. Routine to open the file

```
function Open_File(var FileB : FileVar;  
                   FileName : Str255;  
                   Mode : integer) : integer;  
{  
  This function opens file under Btrieve system and returns  
  code returned from BTrv call.  
}  
var  
  OwnerNameLen : integer;  
  OwnerName : string[128];  
  
begin (Open_File)  
  fillchar(OwnerName, sizeof(OwnerName), 0);  
  OwnerNameLen:=length(OwnerName);  
  fillchar(FileName[length(FileName)+1], 255-length(FileName), 0);  
  Open_File:=BTrv(0, FileB, OwnerName[1],  
                 OwnerNameLen, FileName[1], Mode);  
end; (Open_File)
```

Simply the **BEST** C and Pascal on AT, 386, Sun, Apollo, RT, VAX, 370

"The most rock-solid C compiler in the industry. Superb technical support and portability. Superior code generated."

Gordon Eubanks, Symantec — Q&A (386).

"It simply works, with no trouble, no chasing strange bugs, and excellent warning and error messages ... a professional product."

Robert Lerche, Bay Partners.

"For large-scale software development, the highest quality C compiler available on the market today. Pragmas are great. Quality of support is exceptional." **Randy Neilsen, Ansa—Paradox (DOS, OS/2).**

"15% smaller and 15% faster than Lattice C."

Robert Wenig, Autodesk.

"Our software is running anywhere from 30 to 50% faster than when compiled under Lattice." **David Marcus, Micronetics.**

"We switched from Lattice due to a 10% reduction in code size. The compiler is very stable." **Lee Lorenzen, Ventura Software — Ventura Publisher, marketed by Xerox Corp.**

"Best quality emitted code by any compiler I've encountered. Often amazing." **Bill Ferguson, Fox Software — FoxBase (386).**

"Messages sometimes pointed out type mismatches, incorrect-length argument lists, and uninitialized variables that had been undetected for years [in 4.x bsd]." **Larry Breed, IBM ACIS [RT PC].**

"Diagnostics turned up bugs missed by other compilers. Rapid bug fixes by technical support, someone who knew what he was talking about. 80386 code is well optimized."

Tim Addison, Logistics Data Systems.

"386 protected mode support is fantastic, especially the access to large amounts of memory. It's mainframe compute power on a PC." **Dan Eggleston, Viewlogic.**

"The preprocessor supplied with Professional Pascal is quite useful. The code quality and control over segmentation and memory models are superior to MS Pascal." **Bob Wallace, QuickSoft.**

Check Out These Reviews

• **High C™:**

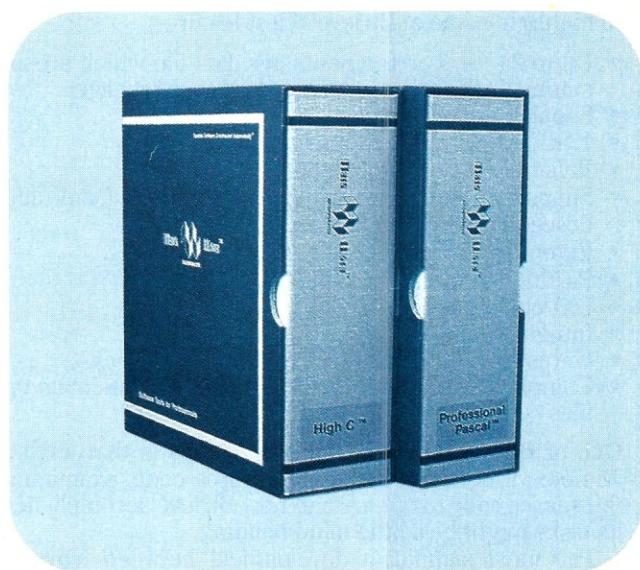
<i>Computer Language</i>	February 1986, '87
<i>Dr. Dobb's Journal</i>	August 1986
<i>PC Magazine</i>	Jan. 27, 1987
<i>Dr. Dobb's Journal</i>	July 1987
<i>BYTE Magazine</i>	November 1987

• **Professional Pascal™:**

<i>PC Magazine</i>	Dec. 29, 1985
<i>Computer Language</i>	May 1986
<i>PC Tech Journal</i>	July 1986
<i>Journal of Pascal, Ada, & Modula-2</i>	Nov.-Dec. 1986
<i>BYTE Magazine</i>	Dec. '86, June '87 (80386 version)

Why MetaWare compilers

- They are specifically designed for serious software developers.
- They are reliable and robust: they don't break at every turn.
- Their generated code is the best, or near best, on each architecture.
- Their superior diagnostic messages help you produce better products more quickly.
- Your source can be ported with ease to the most popular systems.
- You can link mixed-language modules from our compilers, others.
- You can benefit from high-level, personal technical support.
- You can take advantage of the latest ANSI C extensions, and/or extensive Pascal extensions. **High C** has been tracking the ANSI Standard for two years; **Professional Pascal** will soon have a VS Pascal compatibility switch and several Apollo Pascal ext'n's.
- You can take advantage of the latest 387 and Weitek 1167 support — we have the only compilers with Weitek real mode support.



Power Tools for Power Users

Ashton-Tate: dBase III Plus, MultiMate; Autodesk: AUTOCAD; AUTOSKETCH (8087, '387, Weitek); Boeing Computer Services (Sun); CASE Technology (Sun); CAD/CAM giant Daisy ('86, '386, VAX); Deloitte Haskins & Sells; Digital Research: FlexOS; GE; IBM: 4.3/RT, 4680 OS; Lifetree Software (Pascal); Volkswriter Deluxe, GEM-Write; Lugaru: Epsilon; NYU: Ada-Ed cmplr; Semantec: Q&A; Sky Computers; ... (Product names are trademarks of the companies indicated.)

Industrial-Strength

MetaWare C and Pascal compilers are designed for professional software developers. These tools are loaded with options to control them for special purposes. You can adjust the space-time trade-off in code quality. You can adjust external naming conventions to agree with linkers and operating systems. You can specify segment names for segmented architectures, and to help place code or data in particular places for embedded applications. You can select from five memory models for the 8086 family. And on and on.

A Partial List of Optimizations

Common subexpression and dead-code elimination, retention and reuse of register contents, jump-instruction size minimization, tail merging (cross jumping), constant folding, short-circuit evaluation of Boolean expressions, strength reductions, fast procedure calls, automatic mapping of variables to registers (where advantageous), ...

"Platform" — Code Quality

Sun, Apollo, SGI — 18%, 3%, 26% > resident compiler (Dhrystone). PC: DOS, OS/2 — 3-10% > Microsoft C; 30% > MS Pascal, LatticeC. 386 32-bit DOS — no competitors, since November, 1986. 286, 386 UNIX — 66% better than pcc (Dhrystone, 386). VAX VMS — ≈ DEC's excellent C and Pascal; better features. VAX Ultrix — 19% > pcc (Dhrystone); much > Berkeley Pascal. RT PC/4.3bsd — 89% > the original port of pcc (Dhrystone). 370 CMS, UNIX — much better than any C, and VS Pascal. AMD 29000 — >40,000 Dhrystones! Available in Q2, cross.

(408) 429-6382, telex 493-0879.

Since 1979.



903 Pacific Avenue, Suite 201 • Santa Cruz, CA 95060-4429

The Clear Choice for Large Programming Projects

— PC Tech J.

© 1987 MetaWare Incorporated. MetaWare, High C, Professional Pascal, and DOS Helper are trademarks of MetaWare Incorporated. Others and their owners are duly respected.

Some other operations are constructed as shown in Listing 6.

This discussion presents one approach in Pascal; obviously there are different and perhaps better ones. If you are new to Btrieve, however, and to using it with Turbo Pascal, we hope this has provided you with some insight.

Turbo C

The Turbo C approach has evolved somewhat from its Turbo Pascal predecessor. Our C library provides two ways to create a Btrieve datafile. The one shown here is not much different from the Pascal version. The other method uses an outside utility we wrote that creates a special file containing the definition for the Btrieve datafile. When the routine to create the database file is called, it reads in the definition file, then passes this definition to Btrieve for creating the database file. Currently, this is the method we prefer to use. However, the source code for these utilities and routines is too lengthy to include in this article. Those interested can obtain it in our book on Turbo C (*C Developer's Library*, Howard W. Sams & Company).

*From our experience,
we can say
that the DOS
Network Version
of Btrieve
runs like a dream
on Novell networks.*

The C version uses one global arrayed structure rather than several arrays (as seen in the Pascal implementation). This structure is defined as follows:

```
struct
{
    char
        *RecPtr,
        BTrvVar[128],
        CurInd[256];
    int
        RecLen;
} BTrvRecInfo[20];
```

It conveniently stores all the information necessary to the lower-level file manipulation calls and makes important information globally available for more advanced routines. The C equivalent of the Pascal *Make_File* routine to make a Btrieve datafile is shown Listing 7. Implementations of some of the available routines are shown in Listing 8. The first parameter in the *BTrv* call in this listing and the *Pascal* listing is the record manager function request.

Btrieve is by no means the ultimate solution for developing database dependent applications. For one reason, you

only receive 30 days of technical support and the product can be cumbersome to work with until you develop your own methods for cutting through the overhead of learning Btrieve. Once you have familiarized yourself with Btrieve,

Listing 6. Other file function operations

```
function Add_Record(var FileB    : FileVar;
                    FileNum   : integer;
                    var DataRec;
                    CurKeyNum: integer): integer;
{
    This function adds record pointed to by DataRec to file in FileB
}
begin {Add_Record}
    fillchar(CurIndKey[FileNum], sizeof(CurIndKey[FileNum]), 0);
    Add_Record:=BTrv(2, FileB, DataRec, FileRecLen[FileNum],
                    CurIndKey[FileNum][1], CurKeyNum);
end; {Add_Record}

function Delete_Record(var FileB    : FileVar;
                      FileNum   : integer;
                      CurKeyNum: integer): integer;
{
    This function deletes last record retrieved.
}
var
    DumKey: Str255;
    DumRec: char;

begin {Delete_Record}
    fillchar(CurIndKey[FileNum], sizeof(CurIndKey[FileNum]), 0);
    Delete_Record:=BTrv(4, FileB, DumRec, FileRecLen[FileNum],
                        CurIndKey[FileNum][1], CurKeyNum);
end; {Delete_Record}

function Next_Record(var FileB    : FileVar;
                     FileNum   : integer;
                     var DataRec;
                     CurKeyNum: integer): integer;
{
    This function returns next consecutive record unless it cannot
    in which case it will return non zero status referencing reason.
}
begin {Next_Record}
    Next_Record:=BTrv(6, FileB, DataRec, FileRecLen[FileNum],
                      CurIndKey[FileNum][1], CurKeyNum);
end; {Next_Record}

function Search_Record(var FileB    : FileVar;
                      FileNum   : integer;
                      var DataRec;
                      CurKeyNum: integer;
                      var Key     : Str255): integer;
{
    This function puts specified record that matches or is greater
    than Key in DataRec based on path CurKeyNum. If unsuccessful,
    non-zero code indicating reason why will be returned.
}
begin {Search_Record}
    fillchar(Key[length(Key)+1], 255-length(Key), 0);
    Search_Record:=BTrv(9, FileB, DataRec, FileRecLen[FileNum],
                        Key[1], CurKeyNum);
end; {Search_Record}

function Close_File(var FileB    : FileVar):integer;
{
    This function closes Btrieve file and returns Btrieve code from
    BTrv call.
}
var
    DumKeyNum,
    DumRecLen : integer;
    DumDataRec,
    DumKey    : char;

begin {Close_File}
    Close_File:=BTrv(1, FileB, DumDataRec, DumRecLen,
                     DumKey[1], DumKeyNum);
end; {Close_File}

function End_Btrieve : integer;
{
    This function terminates Btrieve, removing record manager and
    closing BTR.TRN file. If unsuccessful returns non-zero
    code indicating reason.
}
var
    DumInt: integer;
    Dummy : Str255;
begin {End_Btrieve}
    End_Btrieve:=BTrv(25, Dummy, Dummy, DumInt, Dummy, 1)
end; {End_Btrieve}
```

C CODE FOR THE PC

source code, of course

<i>NEW!</i>	MS-DOS File Compatibility Package (create, read, & write MS-DOS file systems on non-MS-DOS computers)	\$500
	Bluestreak Plus Communications (two ports, programmer's interface, terminal emulation)	\$400
	PforC or PforCe++ (COM, database, windows, file, user interface, DOS, & CRT)	\$345
	CQL Query System (SQL retrievals plus windows)	\$325
	GraphiC 4.1 (high-resolution, DISSPLA-style scientific plots in color & hardcopy)	\$325
<i>NEW!</i>	Barcode Generator (specify Code 39 (alphanumeric), Interleaved 2 of 5 (numeric), or UPC)	\$300
	Vmem/C (virtual memory manager; least-recently used pager; dynamic expansion of swap file)	\$250
<i>LESS!</i>	PC Curses (Aspen, Software, System V compatible, extensive documentation)	\$250
	Greenleaf Data Windows (windows, menus, data entry, interactive form design)	\$220
	Vitamin C (MacWindows)	\$200
	Greenleaf Communications Library (interrupt mode, modem control, XON-XOFF)	\$175
	TurboTeX (TRIP certified; HP, PS, dot drivers; LaTeX)	\$170
	Essential resident C (TSRify C programs, DOS shared libraries)	\$165
	Greenleaf Functions (296 useful C functions, all DOS services)	\$160
	Essential C Utility Library (400 useful C functions)	\$160
	Essential Communications Library (C functions for RS-232-based communication systems)	\$160
	WKS Library Version 2.0 (C program interface to Lotus 1-2-3, dBase, Supercalc 4, Quattro, & Clipper)	\$155
	OS/88 (U**x-like operating system, many tools, cross-development from MS-DOS)	\$150
	ME Version 2.0 (programmer's editor with C-like macro language by Magma Software; Version 1.31 still \$75)	\$140
	Turbo G Graphics Library (all popular adapters, hidden line removal)	\$135
	PC Curses Package (full Berkeley 4.3, menu and data entry examples)	\$120
	CBTree (B+tree ISAM driver, multiple variable-length keys)	\$115
	Minix Operating System (U**x-like operating system, includes manual)	\$105
	PC/IP (CMU/MIT TCP/IP implementation for PCs)	\$100
	B-Tree Library & ISAM Driver (file system utilities by Softfocus)	\$100
	The Profiler (program execution profile tool)	\$100
	Entelekon C Function Library (screen, graphics, keyboard, string, printer, etc.)	\$100
	Entelekon Power Windows (menus, overlays, messages, alarms, file handling, etc.)	\$100
<i>NEW!</i>	TurboGeometry (library of routines for computational geometry)	\$90
<i>NEW!</i>	QC88 C compiler (ASM output, small model, no longs, floats or bit fields, 80+ function library)	\$90
	Wendin Operating System Construction Kit or PCNX PCVMS O/S Shells	\$80
	C Windows Toolkit (pop-up, pull-down, spreadsheet, CGA/EGA/Hercules)	\$80
	JATE Async Terminal Emulator (includes file transfer and menu subsystem)	\$80
	MultiDOS Plus (DOS-based multitasking, intertask messaging, semaphores)	\$80
	WKS Library Version 1.03 (C program interface to Lotus 1-2-3 program & files)	\$80
	Professional C Windows (lean & mean window and keyboard handler)	\$70
<i>NEW!</i>	lp (flexible printer driver; most popular printers supported)	\$65
	Quincy (interactive C interpreter)	\$60
	EZ_ASM (assembly language macros bridging C and MASM)	\$60
	PTree (parse tree management)	\$60
<i>NEW!</i>	MicroFirm Toolkit (28 Unixesque utilities for MS-DOS)	\$50
<i>NEW!</i>	XT BIOS Kit (roll your own BIOS with this complete set of basic input/output functions for XT's)	\$50
	HELP! (pop-up help system builder)	\$50
	Multi-User BBS (chat, mail, menus, sysop displays; uses Galacticomm modem card)	\$50
	Make (macros, all languages, built-in rules)	\$50
	Vector-to-Raster Conversion (stroke letters & Tektronix 4010 codes to bitmaps)	\$50
	Coder's Prolog (inference engine for use with C programs)	\$45
<i>NEW!</i>	Virtual Memory System (least recently used swapping)	\$40
	C-Notes (pop-up help for C programmers ... add your own notes)	\$40
	Biggerstaff's System Tools (multi-tasking window manager kit)	\$40
	PC-XINU (Comer's XINU operating system for PC)	\$35
	CLIPS (rule-based expert system generator, Version 4.1)	\$35
	Tiny Curses (Berkeley curses package)	\$35
	TELE Kernel or TELE Windows (Ken Berry's multi-tasking kernel & window package)	\$30
<i>NEW!</i>	SP (spelling checker with dictionary and maintenance tools)	\$30
	Clisp (Lisp interpreter with extensive internals documentation)	\$30
	Translate Rules to C (YACC-like function generator for rule-based systems)	\$30
	6-Pack of Editors (six public domain editors for use, study & hacking)	\$30
	Crunch Pack (14 file compression & expansion programs)	\$30
	ICON (string and list processing language, Version 7)	\$25
<i>NEW!</i>	FLEX (fast lexical analyzer generator; new, improved LEX)	\$25
	LEX (lexical analyzer generator; an oldie but a goodie)	\$25
	Bison & PREP (YACC workalike parser generator & attribute grammar preprocessor)	\$25
	AutoTrace (program tracer and memory trasher catcher)	\$25
<i>NEW!</i>	Arrays for C (macro package to ease handling of arrays)	\$25
<i>NEW!</i>	OOPS (collection of handy C++ classes by Keith Gorlen of NIH)	\$20
	C Compiler Torture Test (checks a C compiler against K & R)	\$20
	Benchmark Package (C compiler, PC hardware, and Unix system)	\$20
	TN3270 (remote login to IBM VM/CMS as a 3270 terminal on a 3274 controller)	\$20
	A68 (68000 cross-assembler)	\$20
	List-Pac (C functions for lists, stacks, and queues)	\$20
	XLT Macro Processor (general purpose text translator)	\$20
	C/reativity (Eliza-based notetaker)	\$15
	Data	
	WordCruncher (text retrieval & document analysis program)	\$275
	DNA Sequences (GenBank 52.0 including fast similarity search program)	\$150
	Protein Sequences (5,415 sequences, 1,302,966 residuals, with similarity search program)	\$60
	Webster's Second Dictionary (234,932 words)	\$60
	U. S. Cities (names & longitude/latitude of 32,000 U.S. cities and 6,000 state boundary points)	\$35
	The World Digitized (100,000 longitude/latitude of world country boundaries)	\$30
	KST Fonts (13,200 characters in 139 mixed fonts: specify TeX or bitmap format)	\$30
	USNO Floppy Almanac (high-precision moon, sun, planet & star positions)	\$20
	NBS Hershey Fonts (1,377 stroke characters in 14 fonts)	\$15
	U. S. Map (15,701 points of state boundaries)	\$15

The Austin Code Works

11100 Leafwood Lane

Austin, Texas 78750-3409 USA

Free surface shipping on prepaid orders For delivery in Texas add 7%

acw!info@uunet.uu.net

Voice: (512) 258-0785

BBS: (512) 258-8831

FAX: (512) 258-1342

MasterCard/VISA

you shouldn't need much technical support from Novell. So one piece of advise we have is to jump right into the package and get your questions answered before you have to pay for additional support.

Even with the cost and short support time, Btrieve does have the performance and capabilities for large applications. This makes Btrieve an excellent solution to current developers who need database management that will perform. Best of luck. □

Did you find this article particularly useful?
Circle number 6 on the reader service card.

Listing 7. C routine to make a Btrieve datafile

```
int RMakE_File(char *BTrvVar,
                char *fileName,
                char *fileDef,
                int DefRecLen)
/*
This function will create Btrieve file from definition at
FileDef. Structure at FileDef should be layed out like this:
(This is structure to be used Btrieve)
struct
{
    Char
    Name[31],
    Address[41],
    City[16],
    State[3],
    Zip[11],
    Phone[13];
} Person;

and file creation structure would look like this for
key on Name and Zip fields:
(For purpose of brevity, we will show this structure and
its values as if autoinitilization of structure fields were
possible. However, in practice we must set these values in
executable portion of code.)
struct
{
    int
    RecLen = 115,
    PageSize = 1024,
    NumIndex = 2;
    char
    Blank1[4] = {0,0,0,0};
    int
    Variable = 0;
    char
    Blank2[4] = {0,0,0,0};
    int
    KeyPos0 = 1,
    KeyLen0 = 30,
    KeyFlags0 = 3; (Add together 1 - Duplicates,
                    2 - Modifiable,
                    4 - Binary,
                    8 - Null,
                    16 - Has Segment,
                    32 - Alt Col Seq)
    char
    NU1[10] = {0,0,0,0,0,0,0,0,0,0};

    int
    KeyPos2 = 92,
    KeyLen2 = 11,
    KeyFlags2 = 3;
    char
    NU2[10] = {0,0,0,0,0,0,0,0,0,0};

} BPersonRecDef;
To create and open file as #0, following calls would be made:

RMakE_File(BTrvRecInfo[0].BTrvVar, "Example.Dat",
BPersonRecDef, sizeof(BPersonRecDef));
Open_File(0, "Example.Dat", &Person, 0);
*/
/*Make_File*/
return(BTrv(14, BTrvVar, fileDef, &DefRecLen, fileName, 0));
/*Make_File*/
```

Product Information

Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066
(408) 438-8400
Circle reader service #261

Novell, Inc.
122 East 1700 South
Provo UT 84601
(801) 379-5900
Circle reader service #262

Listing 8. Available C routines

```
int Add_Record(char FileNum,
                char *DataRec,
                int CurKeyNum)
/*
This routine adds data pointed to by DataRec to Btrieve
file (FileNum). Function uses global structure array
BTrvRecInfo[FileNum] to handle miscellaneous record information.
It returns BTrv status code.
*/
/*Add_Record*/
return(BTrv(2,BTrvRecInfo[FileNum].BTrvVar,DataRec,
           &BTrvRecInfo[FileNum].RecLen, BTrvRecInfo[FileNum].CurInd,
           CurKeyNum));
/*Add_Record*/

int Delete_Record(char FileNum,
                  int CurKeyNum)
/*
This function deletes last record (or key) retrieved,
updating necessary information in global structure
BTrvRecInfo[FileNum]. It then returns BTrv status code.
*/
/*Delete_Record*/
char
DumRec;

return(BTrv(4,BTrvRecInfo[FileNum].BTrvVar,&DumRec,&DumRec,
           &DumRec,CurKeyNum));
/*Delete_Record*/

int Next_Record(int FileNum,
                 char *DataRec,
                 int CurKeyNum)
/*
This function places next record in path CurKeyNum
from last one retrieved from FileNum into DataRec. If there
are no more records in search path (it's at end of file),
non-zero status code from BTrv function is returned.
*/
/*Next_Record*/
return(BTrv(6,BTrvRecInfo[FileNum].BTrvVar,DataRec,
           &BTrvRecInfo[FileNum].RecLen, BTrvRecInfo[FileNum].CurInd,
           CurKeyNum));
/*Next_Record*/

int Search_Record(int FileNum,
                   char *DataRec,
                   int CurKeyNum,
                   char *Key)
/*
This function puts specified record that matches or is
greater than Key in DataRec based on path CurKeyNum. It
then returns BTrv status code.
*/
/*Search_Record*/
int
Stat;
setmem(BTrvRecInfo[FileNum].CurInd,
        sizeof(BTrvRecInfo[FileNum].CurInd), 0);
strcpy(BTrvRecInfo[FileNum].CurInd, Key);
Stat = BTrv(9,BTrvRecInfo[FileNum].BTrvVar,DataRec,
            &BTrvRecInfo[FileNum].RecLen,
            BTrvRecInfo[FileNum].CurInd, CurKeyNum);
strcpy(Key, BTrvRecInfo[FileNum].CurInd);
return(Stat);
/*Search_Record*/
```

All the source code for articles published in Micro/Systems is available on an MS-DOS disk. To order, send \$14.95 to Micro/Systems Journal, 501 Galveston Drive, Redwood City, CA 94063; or call Tim at (415) 366-3600. Please specify the issue number. Source code is also available on CompuServe; type GO DDJ and check the data library, line 14.

by P. L. Olympia, Ph.D.

Report Writing in dBASE IV

One of the most glaring weaknesses of dBASE III is its report writing capability. Witness the success of add-on products like R&R Relational Report Writer (Concentric Systems), which makes easy work of creating reports in dBASE and clones. Most people who need elaborate reports tend to write their own from scratch using the dBASE programming language. However, if you're not a programmer, your report requirements tend to be simple, and your forms are of the tabular variety, your best recourse is dBASE's built-in report writer, which is invoked with the CREATE REPORT (or MODIFY REPORT) command. The product of these commands is a .FRM file containing your report specifications in binary form. The report is then produced whenever you give the command REPORT FORM <frm-file>.

Using dBASE III to Create Reports

There are a few problems with the report writer, other than the fact that it was not meant to create very involved reports. For instance, it is difficult to have long report titles and centered headings because you type them into a narrow scrolling window. Also, because the .FRM file is in binary format, it is not easy to document report specifications without having to go back to the report writer and dump the picture of each screen

Dr. P. L. Olympia is a scientist with a doctorate in Chemical Physics. Better known as "Dr. dBASE," he is the author of MEMOPO, a utility program that imports and exports text data to and from dBASE memo fields.

to a file. The thing I dislike most about the report writer, though, is the sheer number of keystrokes required to create even the simplest reports, or to modify an existing report, or to change, say, just the width of one column.

It would be so much easier if you could only design the reports using your own favorite text editor. That would be tantamount to creating a .FRM file with the editor—not exactly a trivial exercise! But what if you could take the report specifications created by a text editor and feed that to a program that could convert those into a .FRM file that dBASE would be unable to distinguish from its own? Such a program could speed up report writing considerably by taking advantage of the editor's search-and-replace functions and block operations. To write such a program, though, you'll need the format of a dBASE .FRM file.

The .FRM File

In the book *dBASE Power: Creating and Using Programming Tools* I explained the format of a .FRM file and showed two relatively simple but useful C programs for handling report writing without dBASE. One of the programs was designed to read a .FRM file and produce an ASCII file that can be changed with a text editor and included in a user manual. The other program was designed to produce a .FRM file from a text file created with an editor. Russ Freeland took the concept one step further when he wrote FRM2PRG, a program that reads a .FRM file and creates a dBASE program from that which can be easily modified and compiled with dBASE

compilers like Quicksilver, Clipper, or FoxBase+. The program and documentation are available from Compuserve and many bulletin boards. Look for a file with the name FRM2PRG.ARC, or something similar.

Space restrictions prevent me from giving a detailed description of the .FRM format here, but a few words about it may be instructive. Unlike other files, a dBASE .FRM file has a fixed length of 1990 bytes. That means dBASE preassigns the space required by the maximum number of columns (24) allowed in a report whether you use all of them or not. The heart of the .FRM file is a 1440-byte area (which I'll call the string heap) meant to contain the character strings used in the report. These include the report title, group and subgroup headings, column labels, and the field expressions defining the contents of each report column. For example, the .FRM file contains pointers to the string heap, where in the heap the report title is stored, and how long a string the title is. The .FRM file uses 12 bytes to describe the particulars of each column, including its width and whether totals should be printed for the column. dBASE reserves space for 25 columns, even though you can only define a maximum of 24 columns in a .FRM file.

Writing Reports In dBASE IV

In dBASE IV, report writing is so much easier and that much more powerful. dBASE IV remains compatible with dBASE III by being able to decode a .FRM file and produce a report from that. You won't really need a .FRM file in dBASE IV since there you can actually design reports simply by painting the screen. When you're done designing your reports, dBASE IV produces a program that can be modified and saved. Run this program any time and you get the report.

Well, if dBASE IV understands a .FRM file created with dBASE III, and writes its report as a .PRG (program) file, couldn't you use it to turn a dBASE III .FRM file into a dBASE III program that you can then embellish with more capabilities? You can, but remember that the resulting .PRG file may need to be revised to translate a few lines with dBASE IV syntax into commands that dBASE III can understand.

Creating a dBASE IV report consists simply of visually specifying what goes into report bands. Basic report bands come in five flavors: Page Header, Report Introduction, Detail, Report Summary, and Page Footer. The Page Header band contains information that appears at the top of each page, such as report date, page

number, or column headings. The Report Introduction band contains information that appears (once) at the beginning of a report; this may consist of word-wrapped text describing the selection criteria and other conditions used to produce the report. The Detail band is the heart of the report. It contains columns of data derived from selected records in the data file. The Page Footer band contains material that appears at the bottom of each page; this may include footnote-type data or information such as the page number, which you are also likely to include in the Page Header band. The Report Summary band contains material that appears at the end of the report; it may contain report totals or word-wrapped text explaining or justifying the report conclusions.

If you want to organize the report based on some grouping criteria, you would also define one or more Group bands. The file must then be indexed (or sorted) on the grouping variable. At the end of each group, you may print descriptive statistics values such as count, mean, and standard deviation.

Printing With dBASE IV

dBASE III knew your printer only as a dumb printer that could print in one direction and occasionally backspace. If you wanted to take advantage of your printer's ability to change pitch, font, print emphasis, print orientation, and the like, you had no choice but to go through a convoluted procedure for embedding printer escape sequences in your .FRM headings and titles. No more. dBASE IV uses printer drivers that let it know automatically the capabilities of your printer. So, you need only say something like:

```
_ppitch = "ELITE"
_pquality = .T.
_pspacing = 2
```

to have the printer change to elite pitch, switch to letter quality mode, and print double space. You may define a default printer driver in the CONFIG.DB file, or change it any time with a command such as:

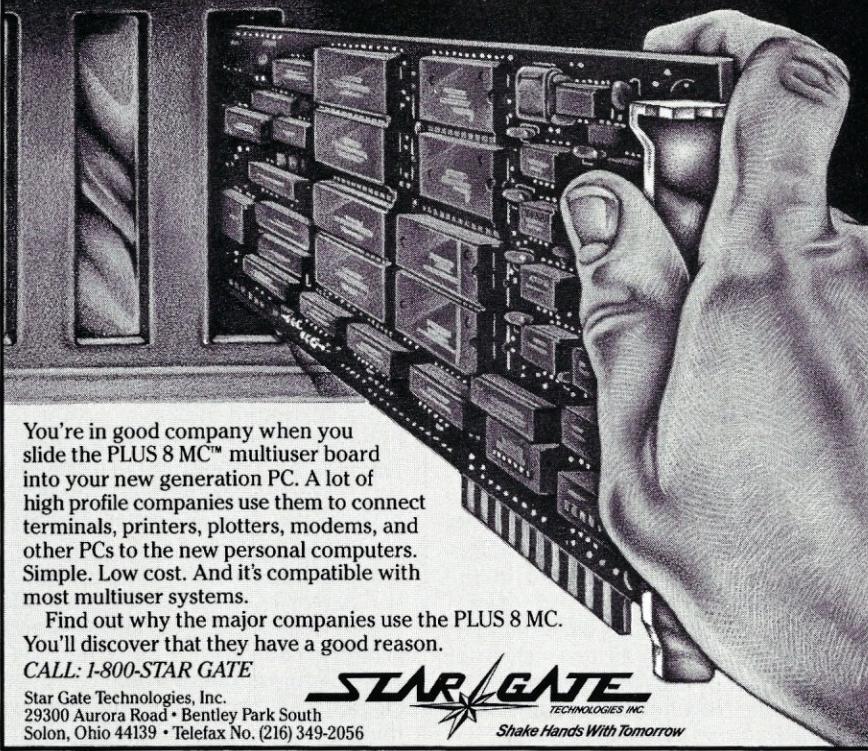
```
_pdriver = "LQ1500_2"
```

which changes the print driver to an Epson LQ 1500. In dBASE IV, only one print driver may be active at a time.

Next time, we'll look at other interesting features of dBASE IV. □

Did you find this article particularly useful?
Circle number 11 on the reader service card.

WE GET INTO THE BEST PLACES



You're in good company when you slide the PLUS 8 MC™ multiuser board into your new generation PC. A lot of high profile companies use them to connect terminals, printers, plotters, modems, and other PCs to the new personal computers. Simple. Low cost. And it's compatible with most multiuser systems.

Find out why the major companies use the PLUS 8 MC. You'll discover that they have a good reason.

CALL: 1-800-STAR GATE

Star Gate Technologies, Inc.
29300 Aurora Road • Bentley Park South
Solon, Ohio 44139 • Telefax No. (216) 349-2056

STAR GATE
TECHNOLOGIES INC.
Shake Hands With Tomorrow

CIRCLE 81 ON READER SERVICE CARD.

PLUS 8 MC is a trademark of Star Gate Technologies Inc.

Disk formats can be a REAL headache!



How many times have you needed to use a file, but your disk was for the wrong computer? Micro Solutions can take the pain out of data transfers by giving you direct access to your Macintosh, IBM PS/2, AT, Apple II, or CP/M diskettes on your PC or AT.

UniForm-PC \$ 64.95
Read, write, copy, and format diskettes for over 275 CP/M and MS-DOS computers on your PC or AT, including 3½", 96 TPI, high density, and 8" formats. Use your DOS programs and commands on your CP/M diskettes.

CompaqCard I \$ 169.95
THE universal four drive floppy controller board for the PC or AT. Runs standard 48 TPI, 96 TPI, & high density 5¼" drives, 720K & 1.44 MB 3½" drives, and 8" floppy drives. Use up to four floppy controllers on the same system. Comes with utility disk. CompaqCard I w/ UniForm-PC program \$ 225.00

MatchPoint-PC \$ 169.95
Use this board and software package with your existing floppy controller to handle NorthStar hard sector and Apple II format diskettes. **Free UniForm-PC included.**

MatchMaker \$ 139.95
Plug in your Macintosh drive to transfer files between Mac and MS-DOS disks. MatchMaker w/ external Mac drive \$ 325.00

Wouldn't it be nice if your PC or AT could speak CP/M?

UniDOS Z80 Coprocessor Board \$ 169.95
Run CP/M programs in your PC or AT with this 8 Mhz. Z80H half-card. UniDOS stays memory resident until a CP/M program is encountered, then transfers execution to the Z80. Terminal emulation built in for Kaypro, Xerox, VT100, and ten others. All standard CP/M system calls are supported. **Bonus: UniDOS and UniForm-PC included.**

30 day money back guarantee
on all products. Please include
\$5.00 for shipping, \$7.50 for COD.
Call or write for full catalog.

**EMERALD
MICROWARE**

Emerald Microware
P.O. Box 1726
Beaverton, OR 97075
(503) 641-0347

CIRCLE 99 ON READER SERVICE CARD.

MICRO/SYSTEMS 63

by A. G. W. Cameron

Should Scientific Computing Use C?

This column is a little bit of a book review, but it is more a discussion of issues raised by that book. The book is *Numerical Recipes in C: The Art of Scientific Computing*, by W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (Cambridge University Press, produced using TeX). The book is a collection of computational algorithms with working expressions of these algorithms in the C language; it is the C version of the book, which in its original form, used FORTRAN and Pascal. I reviewed that version in the July/August 1986 issue of *Micro/Systems Journal*. I also used the FORTRAN subroutines as test cases for translation to C in the review of RTC Plus in the November/December 1987 issue of *Micro/Systems Journal*.

The authors have summed up the premise of the book very nicely in their preface: "This book is unique, we think, in offering, for each topic considered, a certain amount of general discussion, a certain amount of analytical mathematics, a certain amount of discussion of algorithmics, and (most important) actual implementations of these ideas in the form of working computer routines." The scope of the book is to treat everything up to, including a little bit about, partial differential equations. Most collections of scientific subroutines strive for completeness in some sense, but provide little guidance to the conditions in which different algorithms should be used. *Numerical Recipes in C* is the inverse of this practice; the book contains only the more useful algorithms and the authors provide a lot of guidance about how these algorithms should be used. This accounts for the fact that the FORTRAN-Pascal version of the book has become a scientific bestseller.

The C routines described in the book are offered in machine-readable form on diskettes for PCs (\$29.95), and can be obtained from the authors (through Numerical Recipes Software).

A. G. W. Cameron is Professor of Astronomy at the Harvard-Smithsonian Center for Astrophysics

in other formats for mainframes and workstations. There is also a *Numerical Recipes Example Book (C)* (\$19.95) and a *Numerical Recipes Example Diskette (C)* (\$24.95).

C In a FORTRAN World

The authors started working toward the C version of the book as soon as they had finished the FORTRAN/Pascal version. They had been brought up in the usual scientific tradition as FORTRAN users, and to various degrees had experimented with other computer languages. But now they had to learn C, and they had to learn it well enough to program in C and produce code with a style that veteran C programmers would consider elegant. Bill Press is a colleague of mine at Harvard, and starting about three years ago I began to hear comments from him about C being "really weird" and "a crazy language." Such is the culture shock produced when a veteran FORTRAN user is exposed to C.

That culture shock is likely to work both ways in the future. As long as computers were first encountered as research tools in the universities, a student was likely either to learn FORTRAN as an undergraduate, or to enter graduate work with no computing experience. Now that computers are in the public schools and students are coming under the influence of computer science trainees, these students are showing up in the science departments of graduate schools occasionally knowing C but not FORTRAN. My colleagues in the physical sciences thereupon frequently tell these newcomers to forget all that "junk" and learn FORTRAN right away so that they will be able to do something useful. (This is less true of my experimental colleagues, many of whom have learned C as a method of controlling experimental hardware.)

This conflict of computer cultures has been a predictable dilemma. When I first became interested in personal computers, which at the time were CP/M machines, the computer magazines contained frequent denunciations of FORTRAN as being totally

obsolete and very harmful because of its use of "spaghetti code." These sentiments were understandable, but it was quite apparent that the critics had never done serious and substantial scientific calculations, otherwise they would have formed a different opinion about the obsolescence of FORTRAN. Press *et al.* highlight some of these issues. *Numerical Recipes in C* provides, for the first time, a set of scientific subroutine libraries that will enable C users to do serious and substantial scientific computations. But there are lessons to be learned from the way that Press *et al.* had to implement these algorithms for use in the C language. These lessons illustrate why there has been relatively little movement among computational theorists in the physical sciences in the direction of using C for these computations, despite the fact that everyone agrees that the flow of computational control in C is superior.

A major scientific computation usually involves the development of thousands or tens of thousands of lines of code. Following the language revisions that went into FORTRAN '77, these lines of code are considerably less spaghetti-like than was the case for FORTRAN '66, and ongoing revisions to the language will improve things still further. It is almost always the case that the programmer is fighting hardware limitations on storage and/or speed. The traditional C convention is to automatically convert float variables to double before any operation is attempted on them, and then the extra precision is thrown away when the result is returned to a float variable. Press *et al.* characterize this as "sheer madness." This results in a 30 percent to 50 percent loss of speed, and severely degrades the performance of any program for which single precision gives sufficient accuracy. "The practical scientist is trying to solve tomorrow's problem with yesterday's computer; the computer scientist, we think, often has it the other way around," the authors claim.

Another problem is the lack of a C facility that efficiently raises numbers to small integer powers. For example, FORTRAN computes x^4 by multiplying x by itself and then multiplying the product by itself, which is much faster than the procedure for raising a number to an arbitrary power. This may seem relatively unimportant, but remember that scientific computations generally use a lot of polynomials. A FORTRAN expression such as $(A+B)^{**4}$ can thus be done with one add and two multiplies. In FORTRAN compilers it is common to handle

integer powers up to 12 in this way. Press *et al.* sometimes resort to defining common operations, such as squares, as separate functions, but this can sometimes lead to two calls to complicated routines if the argument being squared is such a routine (a trigonometric function, for example). Thus this workaround must be used with caution.

The standard C library function *pow()* is provided for raising numbers to powers, but on many, if not most, implementations, *pow()* requires two float or double arguments. When it allows an integer argument, it frequently just casts to double. And in any case, the language does not formally allow a library routine to have more than one set of types for input.

A Need For C Standards

Press *et al.* point out the need for programming conventions in C. Unlike Pascal, in which the language imposes undesirable constraints, C offers many different ways to do things. Good programming practice, then, is to choose the best and most natural techniques from these possibilities, and to use them in a completely consistent way from one routine to another. For control structures, the authors recommend the use of *for*, *if*, *while*, *do-while*, and *break*. They suggest not using the *switch . . . case . . . default* construction, which is an elaborate form of the FORTRAN "computed goto," because they consider the structure confusing, and also "burdened by uncertainty, from compiler to compiler, about what data types are allowed in its control expression." It is recommended that this be replaced by "a more recognizable and translatable *if . . . else* construction." They also note that the *continue* construction is benign, but that it can usually be replaced by an *if* construction without loss of clarity.

A point of potential confusion concerns the valid range of an index for vectors and one-dimensional arrays. The first element of such an array in C is 0, but in FORTRAN it is 1, and at least as many algorithms used by physical scientists are naturally expressed as going from 1 to M as from 0 to M-1. Press *et al.* wish to liberate the programmer from being tied to the normal C zero-offset conventions, and so they provide a utility file in Appendix D for allocating arbitrary-offset vectors of arbitrary lengths. They also provide corresponding deallocation routines for freeing the allocated vector space. The recipes make extensive use of these utilities.

When we come to matrices and

two-dimensional arrays, a fundamental problem becomes evident. These are what are called *variable dimension arrays* in FORTRAN — arrays that need to be passed to a function together with information about their two-dimensional size. As Press *et al.* note, "The systems programmer rarely deals with two-dimensional arrays, and almost never deals with two-dimensional arrays whose size is variable and known only at run time. Such arrays are, however, the bread and butter of scientific computing. Imagine trying to live with a matrix inversion routine which could work with only one size of matrix!"

There is no intrinsic reason why C should not be able to handle arrays with variable dimensions. However, that is forbidden by the definition of the C language. Press *et al.* are thus forced to adopt a fairly common convention for handling such situations within the confines of C. They avoid altogether any references to the size of a two-dimensional array, which would require specific dimensions. A two-dimensional array *a* is declared by *float *a*; Then the machine code for a specific element *a[i][j]* is in the form "to the address of *a* add *i*, take the value thus addressed as a new address, add *j* to it, return the value addressed by this new address." In this way the underlying size of *a[][]* does not enter at all, and the price paid is the necessary storage for an array of pointers to the rows of the array *a[][]*. One must also remember to initialize those pointers when the array is declared.

The utility provided in Appendix D of the book includes routines to take care of the above. At the same time, provision is made for handling unit offset indices in a manner similar to that used for one-dimensional arrays. The utilities can also be used to set up submatrices and to free up any space that has been allocated.

Don Libes described these matrix-handling techniques in his first *C Forum* for this magazine (*Micro/Systems Journal*, March/April 1985). The trouble is, there are few, if any, elementary books on C language programming that give even a simple example of the techniques. This has undoubtedly inhibited scientific programmers from learning C well enough to discover them.

Much scientific computing is done using complex arithmetic. C does not have a complex data type, nor does it predefine arithmetic operations on complex numbers. Thus Press *et al.* provide a set of utilities in Appendix E to provide these capabilities. The authors caution that the code thus

produced is not very readable, and that they have also had occasion to introduce complex arithmetic operations in a couple of routines using in-line code, and that this is even less readable.

Is C for Science?

From the above discussion it can be seen that the use of C has until now presented a number of problems to the scientific programmer. Press *et al.* have attacked these problems in a variety of ways; some quite satisfactory, others less than fully satisfactory because of the limitations of the language. The book does not attempt to teach the C language to scientific programmers, but it should make it much more practical for them to carry out scientific computations using C.

This book will help computer science types understand why there has been reluctance in the scientific community to abandon the continued use of an "archaic" language like FORTRAN. It makes a case for an expansion of the C language definition so that some of the operations described above can be carried out more gracefully. I understand that current efforts to produce a new C standard are going a little way in this direction, but clearly significantly more improvement is possible. It is also desirable that implementations of the language should not require the sacrifice of execution speed in the run-time code. If this book, and perhaps others like it, lead to the development of substantial libraries of scientific routines in C (at a level of complexity beyond the limits of Press *et al.*), then scientific programmers will probably increasingly do their computations in C. □

Did you find this article particularly useful?
Circle number 10 on the reader service card.

Product Information

Numerical Recipes in C: The Art of Scientific Computing
xxii, plus 735 pages \$44.50

Cambridge University Press
510 North Avenue
New Rochelle, NY 10801
(800) 872-7423
(800) 227-0247 in N.Y.

Numerical Recipes Software
P.O. Box 243
Cambridge, MA 02238

by Phil Hughes

Floppy Disk Data Storage and Transfers

The UNIX operating system allows you to store information on floppy disks in many different ways. Each of these formats has its own uses. In this column, I will explain the different ways to use floppy disks; both as a backup medium and for transferring data between a UNIX system and other computers.

The specific programs I will address are *tar*, *cpio*, and *dd*. All of these programs are part of the standard utilities that come with UNIX. *Cpio* is a relative newcomer to UNIX, but it has been distributed for the last couple of years.

There are three ways to look at the contents of a floppy disk. The data can be a backup medium, foreign data, or a UNIX file system. By backup medium, I mean a place where files can be saved along with their ownership and permission information for purposes of archival storage or backup and restored as desired. The *tar* and *cpio* programs perform these backup functions. The *dump* and *restore* utilities also perform these functions but are designed for complete system or device backups. In this column I will address individual user requirements rather than systems administration.

The *dd* utility is designed to handle foreign data, i.e., data from or to a non-UNIX-based system. It does not address the format of the data but just copies data from one device (or file) to another. It can also perform various conversions, such as ASCII

Phil Hughes is vice president of Specialized Systems Consultants (SSC). In 1983, SSC started offering pocket reference booklets for various UNIX systems and utilities. Since then, SSC has expanded to offer software, training, and consulting to UNIX users.

to EBCDIC, case mappings, and, for magnetic tape, reblocking.

One other capability is using a floppy disk as a mountable file system; in other words, the equivalent of a complete file system with hierarchical directories self-contained on a floppy disk. Unfortunately, the *mount* and *umount* commands, which you must use to allow access to a floppy-based file system, are only available to systems administrators (read that as *root*) on most systems.

Mountable file systems on floppy disks offer all the advantages of a hard disk file system along with the physical and logical security of removable media. These advantages include true random access, file security, and ease of updating. As administrator of the system you might con-

sider using this capability.

Armed with the concepts, let's look at the specifics of each capability. First, backups with *tar* and *cpio*.

Using *tar* and *cpio*

Tar, which stands for tape archive, has been with UNIX since the beginning. It has been the standard archival method for all sizes and flavors of UNIX systems. Common media vary from 360K floppy disks to reel-to-reel, nine-track, half-inch tape. The most common interchange media for UNIX systems is nine-track, 1600 bpi tape. And nine-track reel-to-reel tape drives are available on XT- and AT-class machines. SSC uses one as the primary backup medium for its IBM AT running SCO Xenix.

To use *tar* to create a backup, you invoke it with a list of pathnames on the command line. It then writes all the files named, and the files in specified subdirectories to a specified device. You can then list a table of contents of the *tar* tape or restore files from the tape using the *tar* command with different options. Figure 1 lists the syntax of the command and Listing 1 shows some sample commands. The comments explain their action.

Note that *tar* has keys not options. *Tar* and *ar* are survivors from UNIX of old, back when certain commands had some required control flags. As they were required, no special character was needed to flag them. The first argument following the command name was assumed to be these control flags, called keys.

The *cpio* utility performs much the same function as *tar* but with a different approach. In backup mode, it reads

Figure 1. TAR-Tape file archiver syntax

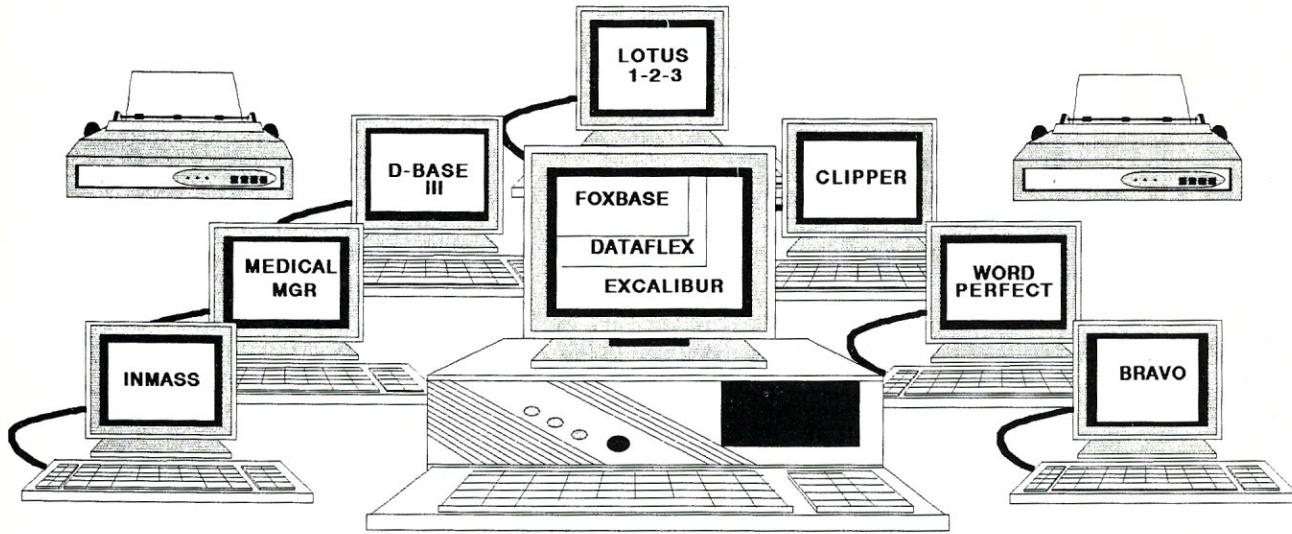
```
% tar [key] [files]
      stdin read if no files specified
Key Format: letter [modifier]

Key Letters:
  c create new tape and record files, implies r
  r record files onto end of tape
  t tell when files found, all entries if no files
  u update tapes by adding files if not on tape or if
    modified since being last written to tape
  x extract files, entire tape if no files

Key Modifiers:
  #      # is tape drive number (0..7), (0 default)
  #den   #density is one of the following:
        h high (6250 bpi)
        l low (800 bpi)
        m medium (1600 bpi) (default)
  b n    n is blocking factor (1 default, 20 max)
  f arch arch is the file to be used for input/output to
    archives (if -then stdin read, /dev/mt? default)
  l      complain if all file links not found
  m      update file modification times
  o      set user and group id of extracted files
        to user running tar
  v      verbose mode
  w      wait for confirmation after reporting filename
        (y causes action to be performed)
```

386-DOS™

A MULTI-TASKING, MULTI-USER CONCURRENT-BASED OPERATING SYSTEM
FOR WYSE MULTI-USER COMPUTER SYSTEMS



386-DOS & SystemWyse—the Multi-user DOS Solution

386-DOS, a Concurrent DOS 386-based operating system from Concurrent Controls, Inc. is a reliable, high performance, multi-user DOS solution for 80386-based microcomputers that combines power and flexibility with high 3-DOS and Concurrent compatibility and virtually unlimited capacity for growth.

386 POWER

Supports up to **16 MB of RAM**
Configure for 1-19 users, printers & devices
Multi-user support retains separate COM1, COM2
19, Televideo 925, Wyse 50 emulation on main console
and terminals, including business graphics, block cursor
Runs most PC-DOS programs on PC-type terminals
Virtual terminal support - 4 on main, 2 at each terminal
Uses intelligent I/O boards for increased performance
Supports large hard disks, **multiple 512MB DOS partitions**
RAMDISK, up to **16 MB** for added speed

PPORT

Complete, professional phone support
Sales, system integrator and programmer support
Provided by the CCI technical support specialists
Ongoing updates and continuing enhancements
Provided for CCI customer base.
Optional BBS support where desired

386-DOS provides a high-performance multiuser DOS solution at a fraction of the costs and implications of a PC network. 386-DOS supports a wide range of DOS applications, including dBase, Foxbase, and DBase III, as well as Concurrent-based packages such as Medical Manager, MASS, Bravo, and Excalibur. Optimized versions provided for many vertical market applications. With 386-DOS your installations can grow from a multitasking, single-user station to large multiuser installation on the same operating system - reducing learning curves, training time and total system costs.

FLEXIBILITY

Use any 8250 based serial port board, or use Wyse 995 Flasher intelligent I/O boards
User-configurable IRQ lines, polled or interrupt driven operation, port buffer sizes, terminal types
Use MS-Net versions of PC-DOS programs
Supports PC-DOS record and file locking
Optional CCIGEN allows on-site OS changes
With CCIXTND, PC-DOS can use all 386-DOS partitions - regardless of size or number
Add users, connect CPUs at any time

EASE OF USE

Familiar PC-DOS interface and commands
Menu-driven installation and maintenance
Single-screen, interactive CONFIG utility sets all serial port parameters and checks for accuracy
Use cost-efficient terminals for additional users - avoid cost and installation of networks



CONCURRENT CONTROLS, INC.

A Digital Research Inc. Master VAR
925A Linden Avenue, South San Francisco, CA 94080
(415) 873-6240 FAX (415) 873-6091

CIRCLE 53 ON READER SERVICE CARD.

Listing 1. Sample tar commands.

```
# backup all files in subdirectory joe
# to the default archive device
tar cv joe
# list table of contents of archive
tar tv
# restore file test in subdirectory joe
tar xv joe/test
# archive all files in the current directory
# to magnetic tape (device rct0)
tar cvf /dev/rct0 .
# extract all files from archive on rct0
tar xvf /dev/rct0
```

Figure 2. CPIO-Copy archives syntax

```
% cpio -i[6bBcdfmrStuv] [patterns]
    copy in: read stdin, select using patterns
% cpio -o[aBcv]
    copy out: pathnames from stdin to stdout
% cpio -p[adlmrv] dir
    copy out and in: read stdin, copy to directory dir
Options:
  6 UNIX 6th edition format
  a reset access times of input files after copy
  b swap halfwords and bytes
  B 5,120 bytes/record (for / dev/ rmt ? only)
  c write header info as ASCII characters
  d create directories as needed
  f copy all files not in patterns
  l link rather than copy whenever possible
  m retain previous file modification time
  r rename files interactively
  s swap bytes
  S swap halfwords
  t print table of contents only
  u copy disregarding age of files
  v verbose, list filenames

patterns      names of files to select,
                specified in shell notation ( * default)
dir          destination pathnames relative to this dir
```

Figure 3. DD-Convert and copy file syntax

```
% dd [options]
Options:
  bs=n  both input and output block sizes
  cbs=n conversion buffer size
  conv=
    ascii  convert EBCDIC to ASCII
    ebcdic convert ASCII to EBCDIC
    ibm   different ASCII to EBCDIC
    lcase map upper case to lower
    noerror continue on error
    swab   swap each pair of bytes
    sync   pad every record to value of ibs
    ucase map lower case to upper
    ..... multiple conversions separated by
          commas
  count=n  copy only n records
  ibs=n  input block size (512 default)
  if=file input filename (stdin default)
  obs=n  output block size (512 default)
  of=file output filename (stdout default)
  seek=n  seek n output records
  skip=n  skip n input records

  n      number of bytes with optional suffix k, b or w
        to specify multiplication by 1024, 512 or 2 or a
        product indicated by two numbers separated
        by an x
```

a list of file names from standard input and writes the archive to standard output. This generally means using a command such as *find* to create the list of file names and redirecting the output to the desired backup device.

In restore mode, *cpio* reads file names from the command line and the archived files from standard input. Wild cards can be used to select files. Figure 2 shows the syntax of the *cpio* command and Listing 2 shows some sample commands. Again, the comments explain the action of the commands.

At first, *cpio* appears harder to use but there are some advantages. First, if the *-c* option is used, the header information is written in ASCII. This

You can write a backup script that only writes out files newer than the last backup.

means that, unless your data files contain binary information, an archive (in other words, a file of files) can be treated like a single ASCII file. You can, for example, use the ASCII mode of a communications program or the UNIX mail program to send this file of files to another system. Using the *-c* option also insures that the header will be understandable on another UNIX system that might represent binary values in a different manner.

Being able to send a list of file names to *cpio* means that you can select the files to be backed up in any manner you wish. For example, using the *-newer* option of *find* you can back up all files newer than a specified file. Using this idea, you can write a backup script that only writes out files that are newer than the last backup. A sample script to do this function is shown in Listing 3.

Listing 3 is a c-shell script but it could have just as easily be written using the Bourne shell. The shell variable *basedir* is set to the top level directory for the backup. This is so you can use relative rather than abso-

lute pathnames with *cpio*. That way you are able to restore files to other than the exact path from which they were backed up.

The shell variable *dlist* is initialized to a list of all the subdirectories of *basedir* that is to be included in the backup. *outdev* is set to the name of the backup device, in this case magnetic tape unit 0.

The shell variables *day* and *all* are initialized to the names of the control files that will be used to keep track of when the last dump was performed. *day* is for the incremental dump and *all* is for the full dump. The next few lines prompt for the dump type and validate the user input.

The actual dump is performed by setting the current directory to *basedir* and running *find* to supply the file names to *cpio*. The first *find* command prints out all files in the specified subdirectories and, thus, *cpio* will back up all files. The second *find* command is the one run if an incremental dump was requested. It uses the *-newer* option of *find* that selects all files newer than a specified file. The file specified is the daily control file; the name we saved in the *day* shell variable.

Finally, if the dump was successful, the *touch* command is used to update the last modify time of the appropriate control file. This is done last to insure that the control file is only updated upon successful completion of the dump. This does leave a loophole, however. Any file modified after the dump was started but before it was completed may not get dumped. Since you should run a dump on an idle system to insure the integrity of the files being dumped, this problem should not occur.

Using *dd*

dd is another established UNIX utility. Its command syntax differs from the norm. Each option is expressed as *option = value*. For example, an input file named *stuff* would be expressed as *if=stuff*. Figure 3 shows the complete syntax for the *dd* command. Listing 4 shows some common usage for *dd*. □

Did you find this article particularly useful?
Circle number 12 on the reader service card.

The figures presented in this column are extracted from SSC's UNIX System5 Command Summary, Copyright 1987, SSC, Inc. Used with permission.

Listing 2. Sample *cpio* commands.

```
# backup all files whose names are in flist
# to device fd0
cpio -oc <flist >/dev/fd0
# list table of contents of previous backup
cpio -itv </dev/fd0
# backup all files in current directory and subdirectories
# output to device mt0
find . -print | cpio -oc >/dev/mt0
# restore all files and directories from tape
cpio -icd </dev/mt0
# as above but preserve old modify time of files
cpio -icdm </dev/mt0
```

Listing 3. Sample script file to write out files newer than last backup.

```
# sysbak -- backup files -- ph - 11-20-87
# backup a selected list of files
# Prompts for a full or incremental dump

# top of the hierarchy to consider for dumping
set basedir = /u
# which directories to dump
set dist = (alex control dlc4 local zbs)
# output device for dump
set outdev = /dev/rmt0
# file names for each control file
set day = /u/control/day          # incremental
set all = /u/control/epoch        # full

echo "enter dump type: a for all or d for daily"
set type = 'gets'
switch ($type)
  case "d":
    echo "Daily dump"
    breaksw
  case "a":
    echo "Full dump"
    breaksw
  default:
    echo "sysback: invalid option"
    exit
    breaksw
endsw

# do the dumps
echo "Load tape and hit <return>"
gets           # wait for return

cd $basedir    # so we can use relative pathnames
if ($type == "a") then      # full dump
  find $dlist -print | cpio -oBvac > $outdev
else
  find $dlist -newer $day -print | cpio -oBvac > $outdev
endif

if ($status != 0) then
  echo "****BACKUP FAILED****"
else
  # update dump info
  if ($type == "a")
    touch $all
  else
    touch $day
  end
endif
```

Listing 4. Using the *dd* command.

```
# read contents of tape mt0 into file mine
# converting upper case characters to lower case
# and EBCDIC to ASCII
dd if=/dev/mt0 of=mine conv=lcase,ascii
# read a floppy disk into file real.file on the hard disk
# all of the floppy will be read
dd if=/dev/fd0 of=real.file
# as above but larger block size will make it run faster
# 15b should be a track on a 1.2MB floppy
dd if=/dev/fd0 of=real.file bs=15b
```

New Products

New Software Products

SQL Development Tools Support Multiple Database Servers

Gupta Technologies has introduced SQLWindows, a database application development system to help programmers develop applications for the Microsoft Windows environment without using C programming or the Windows toolkit. SQLWindows applications will operate on any 286/386 workstation running Windows 2.0 or Windows/386. It includes a graphics forms editor for laying out the visual presentation of a form, including fields, buttons, scroll bars, menus, and pictures. Actions are coded in an internal procedural language called SAL (SQLWindows Applications Language) and can contain embedded SQL statements to access a database. Gupta has also released an OS/2 version of its SQLBase Server that also serves as a back end for SQLWindows. SQLBase Server can be used as a multiuser server or as a single-user DBMS engine. An OS/2 Presentation Manager version of SQLWindows is being developed.

The SQLWindows Systems Development Kit (SDK) is available for \$1,295 and contains the Design and Run-Time modules, the SQR Report Writer, and the SQLTalk Interactive Data Manager. It also includes SQLBase/16M, a version of SQLBase that runs above the 640K base memory for single users. The SQLWindows Application Design Module is available separately for \$595 per workstation. The SQLBase SDK is available for \$2,995 for the multiuser version, and \$1,295 for the single-user version. For more information, contact **Gupta Technologies, Inc.**, 1020 Marsh Road, Suite 210, Menlo Park, CA 94025; (415) 321-9500.

Circle reader service #15.

Quadbase DBMS Now Supports SQL/QBE

Quadbase Systems has released dQUERY 2.1 and dQUERY/Lib 2.1, which support Lotus 1-2-3 file formats and dBASE III file formats. dQUERY is an *ad hoc* query management sys-

tem and report writer that consists of a relational database engine, a Structured Query Language/Query-By-Example interface, and a report writer to query dBASE and 1-2-3 files, manipulate data, and generate reports. It features a multiple window environment with pull-down menus and extensive function keys. The QBE interface allows users to make queries by making entries in forms drawn on the screen. With version 2.1, dQUERY can now query on Lotus 1-2-3 files, transparently join 1-2-3 and dBASE III files, support the SQL "alter table" statement, support arithmetics, import ASCII files, and open up to 32 database files simultaneously. dQUERY/Lib 2.1 features a set of Microsoft-C callable routines that allow software developers to gain access to dQUERY's relational database engine.

dQUERY 2.1 requires 512K of RAM and is priced at \$150. A batch "runtime" version is available at no extra cost. dQUERY/Lib 2.1 is priced at \$295. For more information, contact **Quadbase Systems Inc.**, 790 Lucerne Drive, Suite 51, Sunnyvale, CA 94086; (408) 738-6989.

Circle reader service #16

NetWare Now Has SQL Engine

The Development Products Division (DPD) of Novell has introduced NetWare SQL, a Structured Query Language engine optimized for NetWare. Based on NetWare and Btrieve, NetWare SQL offers many of the benefits of SQL, including transaction- and user-access-level security using passwords, trustee privileges, and standard backup utilities and fault tolerance through SFT NetWare. Through XQL, NetWare SQL provides a powerful API for software developers. XQL provides multiple language interfaces, a variety of data types, and application-defined buffering. The initial release of NetWare SQL will support DOS and OS/2. In a related announcement, Borland International, Aker Corporation (maker of Magic PC), Advanced Business Microsystems, Inc. (makers of Platinum), and Zanthe Information Inc. (maker of ZIM) have pledged to support NetWare SQL as a back end to their applications. Further, with support from Lotus Devel-

opment Corporation, Novell will develop a Lotus Blueprint driver that will enable NetWare SQL to work with Lotus products.

NetWare SQL is expected to become available in the fourth quarter and will be priced at \$595 per server. DPD will also offer telephone support services for a nominal fee. For more information, contact **Novell, Inc.**, 122 East 1700 South, Provo, UT 84601; (801) 379-5900, or contact your Novell distributor.

Circle reader service #17

New Data Acquisition Package for DOS

Remote Measurement Systems, Inc., has released the ADC-1 Applications Toolkit. This MS-DOS package enables both seasoned and novice programmers to create data acquisition and measurement programs. The Toolkit includes sample programs to aid in the design of data collection programs and enhances the company's ADC-1 Data Acquisition and Control System. It also provides source code in BASIC, C, Pascal, dBASE III+, and Quicksilver. It includes full graphics support for the PS/2 using QuickBasic 4.0, and supports color monitors with programs written in BASIC.

The ADC-1 Applications Toolkit retails for \$45. For more information, contact **Remote Measurement Systems, Inc.**, 2633 Eastlake Avenue, Suite 206, Seattle, WA 98102; (206) 328-2555.

Circle reader service #18

New Hardware Products

Peripherals Offer Easy Upgrade to PS/2

A new line of PS/2 enhancements has been introduced by NeoTech, Inc., encompassing interface adapters, MicroChannel cards, and associated software. First in the lineup is an external 5 1/4-inch disk drive. NeoTech's MicroChannel Parallel Adapter is fully bidirectional. Also included is a MicroChannel Serial Adapter for interfacing to RS-232 peripherals. The serial board can operate at 19.2K baud and higher. Both the parallel and serial boards comply with IBM specifications.

The 5 1/4-inch disk drive sells for \$295. The Parallel Adapter retails for \$99.95 and Serial Adapter sells for \$129.95. For more information, contact **NeoTech, Inc.**, 30295 Solon Industrial Parkway, Solon, OH 44139; (216) 248-4114.

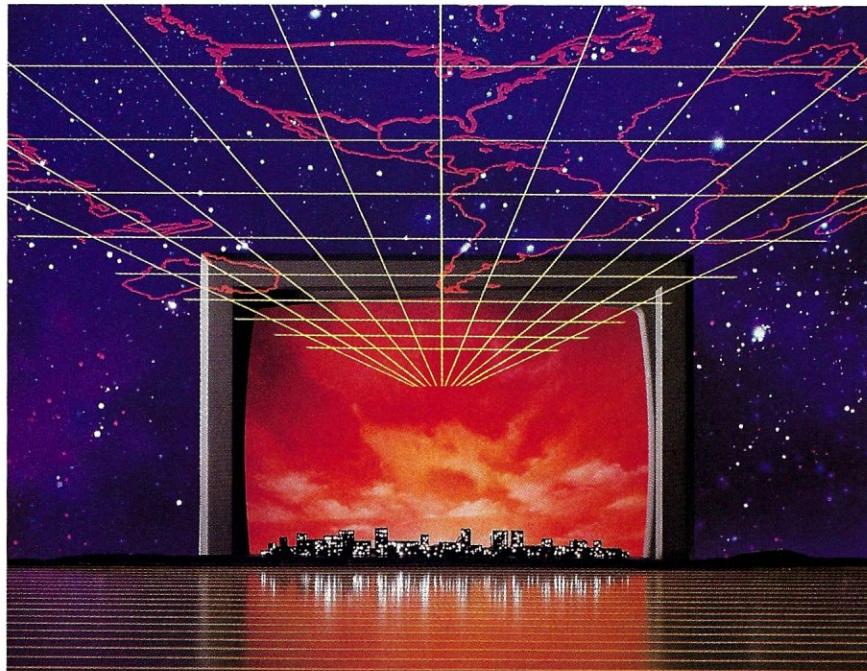
Circle reader service #19

(continued on page 72)



THE UNIX™ OPERATING SYSTEM EXPOSITION & CONFERENCE

OCTOBER 31 - NOVEMBER 2, 1988
JACOB K. JAVITS CONVENTION CENTER
NEW YORK CITY



PHOTOGRAPH: MICHAEL TCHEREKOFF

YOUR GUIDE TO EFFECTIVE SOLUTIONS

Recent developments have accelerated the acceptance of UNIX system applications and solutions as a viable option for business. As a result, it is now increasingly vital that business and computer professionals be familiar with new developments in hardware, software, standards, connectivity, open systems and other issues that impact on how you invest your time and money.

With UNIX EXPO as your guide, you will be able to effectively use the power of the UNIX system. At UNIX EXPO, explore new ways of enhancing and expanding your current installation while protecting your previous investments. The outstanding solutions widely available with multi-user technology at UNIX EXPO will be invaluable to current UNIX users as well as those new to the UNIX system.

Benefit from more than 150 leading manufacturers and suppliers exhibiting at UNIX EXPO and

to explore first hand, where you should be investing in your company's future.

The highly regarded Conference is an outstanding opportunity to have industry experts and experienced UNIX system users share their experience and insight with you. The tutorials, presented by AT&T, provides you with concrete, practical information that can immediately be applied to benefit your organization.

Attend UNIX EXPO, where business users, resellers and systems integrators will gain the knowledge, the insight and the solutions to meet the challenge of increased productivity and profitability. Make your plans today and let UNIX EXPO be your guide.

For more information on UNIX EXPO, contact: National Expositions Company, Inc., 15 West 39th Street, New York, NY 10018, USA, Tel: 212/391-9111 Fax: 212/819-0755 Telex: 135401 dimcomm



Boca Adds PS/2 I/O Board

Boca Research has begun shipping the BOCA.MCA Parallel, a new input/output adapter for the PS/2 Micro-Channel. The board supports the MicroChannel's shared interrupt scheme and is configurable to all PS/2 base addresses. Each 25-pin parallel port is addressable as LPT1 through LPT3. The board can be connected to either a 16-bit or 32-bit MicroChannel slot and can be installed using the IBM setup program and the .ADF file supplied with the board. It measures 11.5 by 3.475 inches and uses 0.25 amps at 5 volts.

The BOCA.MCA Parallel board sells for \$110. For more information, con-

tact **Boca Research, Inc.**, 6401 Congress Avenue, Boca Raton, FL 33487; (407) 997-6227.

Circle reader service #20

Rewritable Optical Introduced by AGA

Advanced Graphics Applications (AGA) has unveiled DISCUS Rewritable, the first "plug and play" 5 1/4-inch rewritable optical disk system for MS-DOS. With DISCUS (Data, Image, Sound, Communications, Unified Storage), PC users can store, retrieve, modify, and delete data on 650-MB rewritable optical disks. The DISCUS Rewritable can be connected to a stand-alone workstation or a network file

server. DOS compatibility makes it possible to exchange information between rewritable optical, WORM, CD-ROM, Winchester, and floppy disks. Unlike other drives, however, the media is removable with a high storage capacity. The device driver software will also comply with the ISO standard, the same standard as Matsushita, Maxtor, Olympus, and Sony equipment. By November, company executives expect to have an OS/2 device driver ready for market. Device drivers for UNIX and Macintosh operating environments are being developed as well. The DISCUS Rewritable can be configured like any peripheral: AGA's AGA DRIVE SCSI Host Adapter is installed in a slot in the PC, and the device driver software is loaded using an additional line in CONFIG.SYS.

The DISCUS Rewritable is available for \$4,995. 3M is manufacturing the 650MB erasable optical disks, which are available for \$250 from dealers. For more information, contact **Advanced Graphics Applications**, 90 Fifth Avenue, New York, NY 10011; (212) 337-4200.

Circle reader service #21

Improving Connectivity from DOS to VAX

Datamedia Corporation has released a new generation of its Colorsan/2 graphics workstation. The new Colorsan/2 offers a full 640K DOS session while simultaneously running a ROM-based VT220 terminal session. Users simply "hot key" between sessions and transfer information between VAX/VMS and DOS applications. The Colorsan/2 VT220 terminal emulator is entirely ROM based and does not use any of DOS's 640K memory. LAN support is also offered using Digital Equipment's local area transport (LAT) that allows the workstation to operate as a VT241-compatible ReGIS graphics terminal over Ethernet networks. Combining LAN support with the DECnet PCSA lets Colorsan/2 use a single Ethernet connection to run a DOS session and a VAX/VMS session simultaneously. Novell NetWare boot software is also available on the Colorsan/2 CARDfile, a "smart card" device that provides read-only software for a diskless Colorsan/2 with a Datamedia Ethernet Adapter and drivers to boot MS-DOS from a NetWare file server. An EMS driver is also included, along with a printer hot key, and DEC printer support.

For more information, contact **Datamedia Corporation**, 11 Trafalgar Square, Nashua, NH 03063; (603) 886-1570.

Circle reader service #22

OMNIVIEW ROCK SOLID MULTITASKING!

Packed with the power you need, OMNIVIEW takes you beyond the limits of DOS.

- Time-slices multiple jobs
- Takes full advantage of EMS 4.0
- Optional support for '386 Systems
- Manages megabytes of programs
- Controls resident utilities
- Lets you cut and paste between programs

OMNIVIEW unleashes your PC & converts it into an easy-to-use, multitasking workstation. Whether you have an 80386 or 8088 based computer, OMNIVIEW will squeeze out every ounce of power.

Solid & bulletproof, it's 47K assembly language engine gives you all the power without added overhead or complexity. Just ask the reviewers.*

"Sunny Hill provides true multitasking for most PC applications. It performs remarkably well, causing almost no noticeable reduction in the performance of each program... The user manual is well organized and easy to read... It is an excellent product."

PC Tech Journal Sept '87
"It does an amazing job of handling not only multitasking, but in many cases, concurrent processing as well. I was amazed at its success in handling the 'impossible' combinations I threw at it."

"It is one of the best designed & implemented programs I've ever worked with. It is also one of the most bulletproof."

"What sets it apart from the pack is the fact that it runs in a network environment."

Data Based Advisor Aug '87

OMNIVIEW requires an IBM PC, XT, AT or Jr compatible, and PC or MS DOS 2.x to 3.x. To get your copy, call toll free:

(800) 367-0651

or send \$89.95 + 3.00 S&H (\$8.00 Int'l.) to:



Sunny Hill Software

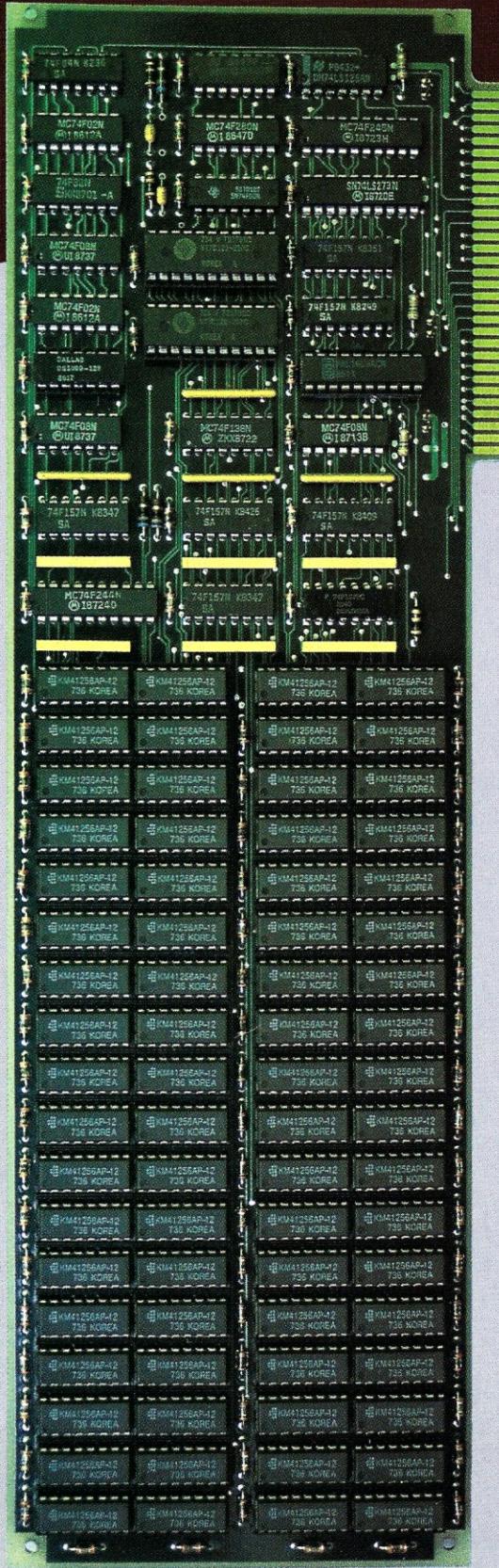
PO Box 55278, Seattle, WA 98155-5278

For more information call (206) 367-0650 *Omniview formerly Taskview
DoubleDOS trademark Softlogic Solutions, Concurrent PC-DOS reg. trademark Digital Research Corp., MS Windows
trademark Microsoft Corp., DESQview trademark Quarterdeck Office Systems, Inc. TopView trademark IBM Corp.

CIRCLE 51 ON READER SERVICE CARD.

X-BANDIT

Designed for the EMS 4.0 Standard



DESIGN PHILOSOPHY

- The Teletek X-Bandit was specifically designed to utilize the advanced features of the Lotus/Intel/Microsoft EMS 4.0 Specification. It is available in both 8 and 16 bit versions for use in the IBM XT, AT, and compatibles.

MEMORY

- Segmented Memory Mapping allows the user to fill out unused memory segments between 640K and 1 Megabyte.
- Split Memory Addressing allows the user to fill out conventional memory to 640K.
- Extended Memory Addressing is available for the PC/AT version.
- 2 MB capacity in a single slot. Up to 8 MB per system.
- Parity checking.

SOFTWARE

- Easy menu-driven auto configuration software.
- Device driver includes print spooler and RAM drive.
- Supports multitasking with the appropriate shell-resident software package.

SPEED

- 6/8/10/12 MHz speed with 0 wait states. 16 MHz speed with 1 wait state.

WARRANTY

- One year parts and labor.

TELETEK

4600 Pell Drive
Sacramento, CA 95838
(916) 920-4600
Telex #499-1834



DOS system running Lotus 1-2-3

THIS IS AN IBM PS/2 MODEL 80 RUNNING DOS

Under DOS, this PS/2™ is a powerful 80386-based single-tasking, single-user computer that can run thousands of DOS applications. In 16-bit, 8086 mode.

One at a time.

When OS/2™ software becomes available, the PS/2 can become a multitasking, single-user computer running in 16-bit, 286 mode that can also single-task those DOS applications under OS/2.

One at a time.

With DOS or OS/2, the PS/2 will support one user.

	1 user (DOS)	1 user (OS/2)
Cost per system**:	\$12,389	\$12,594
Cost per user:	\$12,389	\$12,594



SCO XENIX system running SCO Professional

THIS IS AN IBM PS/2 MODEL 80 RUNNING SCO XENIX

Under SCO XENIX®, this PS/2 becomes a powerful 80386-based multitasking, multiuser computer that can run thousands of XENIX applications. In full-tilt, 32-bit, 386 mode.

Many at a time.

And using SCO VP/ix™*, the PS/2 can multitask DOS applications under SCO XENIX.

Many at a time.

With SCO XENIX, the PS/2 will support one user. Or 9 users. Or even 33 users.

And it can do all that today because you can get SCO XENIX for the PS/2—*now!*

	1 user	9 users	33 users
Cost per system**:	\$14,559	\$19,726	\$40,402
Cost per user:	\$14,559	\$2,192	\$1,224

SCO XENIX System V and the SCO XENIX family of software solutions are available for all industry-standard 8086-, 80286-, and 80386-based computers, and the IBM® Personal System/2™ Models 50, 60, and 80.



(800) 626-UNIX (626-8649)

(408) 425-7222

FAX: (408) 458-4227

TWX: 910-598-4510 SCO SACZ

uucp: ...decvax!microsoft!SCO!info!

*SCO VP/ix available as separate product.

**Cost comparisons are based on most recently published U.S. domestic suggested list prices. Cost model: Base machine: IBM PS/2 Model 80, 70MB disk, 1Mb RAM, IBM 8512 color monitor, 1Mb additional IBM RAM, IBM ProPrinter XL®. 1-user DOS system: Base machine, plus SCO XENIX 386 for PS/2 DOS. 1-user SCO XENIX system: Base machine, plus SCO XENIX 386 for PS/2, SCO VP/ix, SCO Lynix® (word processing), SCO FoxBASE+™ (dBASE III PLUS workalike), SCO Professional™ (1-2-3 workalike). 9-user SCO XENIX system: 1-user SCO XENIX system, plus intelligent 8-user multiport card, 8 IBM 3151 ASCII terminals. 33-user SCO XENIX system: 9-user SCO XENIX System, plus 3 more intelligent 8-user multiport cards, 24 more IBM 3151 ASCII terminals, 4 Mb additional RAM, additional 70MB disk.

IBM and ProPrinter XL are registered trademarks and Personal System/2, PS/2 and OS/2 are trademarks of International Business Machines Corporation. • Lotus and 1-2-3 are registered trademarks of Lotus Development Corporation. • dBASE III PLUS is a registered trademark of Ashton-Tate. XENIX is a registered trademark of Microsoft Corporation. • VP/ix is a trademark of INTERACTIVE SYSTEMS, Inc. • Lynix is a registered trademark and SCO Professional is a trademark of The Santa Cruz Operation, Inc. • FoxBASE+™ is a trademark of Fox Software, Inc. 10/87 The Santa Cruz Operation, Ltd., 18 Noel Street, P.O. Box 41N, London W1A 4N United Kingdom; +44 1 439 2911, (FAX): +44 1 637 9381, TELX: 917372 SCO.LOV

CIRCLE 94 ON READER SERVICE CARD